



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ZAJIŠTĚNÍ PŘESNOSTI
STEREOZPRACOVÁNÍ OBRAZU**

ENSURING PRECISION OF STEREO IMAGE PROCESSING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

OLIVER KUNÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2021

Zadání bakalářské práce



23811

Student: **Kuník Oliver**

Program: Informační technologie

Název: **Zajištění přesnosti stereozpracování obrazu**
Ensuring Precision of Stereo Image Processing

Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte dostupnou literaturu na téma metod stereo zpracování obrazu a jejich přesnosti včetně kalibračních postupů.
2. Navrhněte metodu zpřesnění a vyhodnocení přesnosti dosažitelné stereo metodami při měření polohy a/nebo rychlosti na základě kompenzace pohybu kamer.
3. Vyberte vhodnou metodu vyhodnocení a zhodnoťte dosažitelné vlastnosti měření.
4. Implementujte vybranou metodu a demonstруйте funkčnost na vhodném příkladu.
5. Diskutujte výsledky a možnosti pokračování v práci.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Táto práca sa zaoberá možnosťami spresnenia výsledkov spracovania stereoobrazu a skúma spôsoby jeho stabilizácie. Navrhuje postup stabilizácie stereozáznamu vzhľadom na určenie referenčnú stereosnímku pre udržanie snímok v skalibrovannej polohe kamery. Na eliminovanie pohybu stereosnímkov je použitá afinná transformácia. Na jej hľadanie sú použité pozície rohových bodov nájdené so subpixelovou presnosťou. Práca obsahuje porovnanie účinnosti stabilizácie stereozáznamu určeného na počítanie rýchlosti vozidiel.

Abstract

This thesis deals with the possibilities of refining the results of stereo image processing and examines the methods of stereo image stabilization. Proposes a method of stabilizing the stereo recording with respect to the designated reference stereo image to keep the images in the calibration position of the camera. An affine transformation is used to eliminate the movement of stereo images. The positions of the corner points found with subpixel precision are used for calculating the transformation. This thesis includes a comparison of the stabilization efficiency of a stereo record intended for vehicle speed measurements.

Klíčové slová

stabilizácia snímok, stereokamera, afinná transformácia, Shi-Tomasi detektor, subpixelová presnosť, vzájomná korelácia, meranie rýchlosti, OpenCV

Keywords

image stabilisation, stereo camera, affine transformation, Shi-Tomasi detector, subpixel precision, cross-correlation, speed measurement, OpenCV

Citácia

KUNÍK, Oliver. *Zajištění přesnosti stereozpracování obrazu*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Dr. Ing. Pavel Zemčík

Zajištění přesnosti stereozpracování obrazu

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána prof. Dr. Ing. Pavla Zemčíka. Ďalšie informácie mi poskytol pán Ing. Pavel Najman. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Oliver Kuník

12. mája 2021

Podakovanie

Rád by som poďakoval svojmu vedúcemu bakalárskej práce, pánovi prof. Dr. Ing. Pavlovi Zemčíkovi za vedenie a konzultácie pri vypracovávaní bakalárskej práce. Tiež by som chcel poďakovať pánovi Ing. Pavlovi Najmanovi za konzultácie a pomoc pri testovaní.

Obsah

1	Úvod	2
2	Stereovidenie	3
2.1	Počítačové videnie, obraz a kamera	3
2.2	Stereokamera a stereovidenie	6
2.3	Existujúce kalibračné postupy a ich presnosť	8
2.4	Existujúce práce zaoberajúce sa presnosťou stereokamier	10
3	Algoritmy spracovania obrazu	13
3.1	Hľadanie vhodných bodov na sledovanie	13
3.2	Zisťovanie podobnosti rôznych bodov	17
3.3	Transformácie obrazu	18
3.4	Párovanie korešpondujúcich bodov	20
3.5	Prevzorkovanie obrazu	21
4	Zhodnotenie problému a návrh parametrov riešenia	22
4.1	Nepresnosť meraní spôsobená pohybom kamery	22
4.2	Návrh parametrov, ktorými by sa mal kompenzačný algoritmus riadiť . . .	23
5	Implementácia navrhnutého riešenia	25
5.1	Spracovanie referenčnej snímky	26
5.2	Rýchly algoritmus na hľadanie korešpondujúcich rohov	28
5.3	Robustnejší algoritmus hľadania korešpondujúcich rohov	29
5.4	Výpočet transformačnej matice afinnej transformácie	30
6	Testovanie implementovaného riešenia	31
6.1	Hľadanie optimálnych parametrov	31
6.2	Problémy odhalené počas testovania	33
6.3	Testovanie reálnej funkčnosti algoritmu	35
7	Záver	39
	Literatúra	41

Kapitola 1

Úvod

S príchodom digitálnych fotoaparátov a kamier vznikol nový odbor informačných technológií, počítačové videnie. Počítačové videnie sa zaoberá získavaním informácií z obrazu. Dnes sa počítačové videnie využíva v mnohých situáciach, či už ide o ovládanie robotov a automatizovaných vozidiel, vytváranie 3D modelov z reálnych objektov alebo sledovanie dopravy. V niektorých oblastiach, ako napríklad pri meraní rýchlostí vozidiel je potrebné, aby bola zaručená presnosť týchto meraní. Práve zaisteniu presnosti stereospracovania obrazu sa venuje táto práca.

Zaručenie presnosti je dôležité. Napríklad systém kontrolujúci rýchlosť vozidiel prechádzajúcich po komunikácií, ktorý meria s odchýlkou ± 10 km/h, sa nedá využiť na udeľovanie pokút. Aby takéto merania boli čo najpresnejšie, vyvíjajú sa stále lepšie a presnejšie metódy na kalibráciu kamier. Avšak na jeden problém sa často zabúda. Aj keby bola kamera perfektne skalibrovaná, nie je zaručené, že bude merať presne, pretože dopravná kamera sa hýbe. Aj keď je pripevnená o oceľový profil mosta, stále je ovplyvnená vonkajšími vplyvmi ako vietor alebo prechod ťažkého nákladného automobilu, ktoré dokážu rozvibrovať most alebo inú konštrukciu, na ktorej je umiestnená kamera. Tým sa rozpohybuje aj samotná kamera. Tieto posuny potom môžu zhoršiť presnosť nameraných výsledkov, keďže kamera je skalibrovaná pre nejakú pozíciu, v ktorej sa v danom momente nemusí nachádzať. Táto práca sa zaoberá implementáciou a testovaním metódy, ktorá tieto pohyby detekuje a následne kompenzuje.

V nasledujúcej kapitole je vysvetlené, čo je to počítačové stereovidenie, ako funguje a ako sa v súčasnosti zaručuje jeho presnosť. V tretej kapitole sú popísané algoritmy užitočné pri hľadaní transformácie medzi dvomi obrazmi. Štvrtá kapitola rozoberá problém pohybu kamery a predstavuje návrh riešenia tohto problému. V piatej kapitole je popísaná implementácia riešenia a v šiestej kapitole sú predstavené výsledky z testovania a zhodnotenie funkčnosti naimplementovaného riešenia.

Kapitola 2

Stereovidenie

Táto kapitola sa zaoberá tým, ako funguje počítačové stereovidenie a prečo je nutná presnosť pri stereospracovaní obrazu. Nejedná sa o encyklopedický prehľad, uvedené sú tu len informácie, ktoré súvisia s touto prácou. Prvá časť kapitoly sa zaoberá tým, čo je to počítačové videnie a vďaka akým prístrojom a princípom môže počítač „vidieť“. Druhá časť je zameraná na stereokameru a stereovidenie. Rozoberá, akú výhodu ponúka oproti klasickému monovideniu, na akom princípe funguje a aké problémy je potrebné pri stereovidení riešiť. Posledná časť je zameraná na existujúce riešenia problémov spomenutých v prvej a druhej časti.

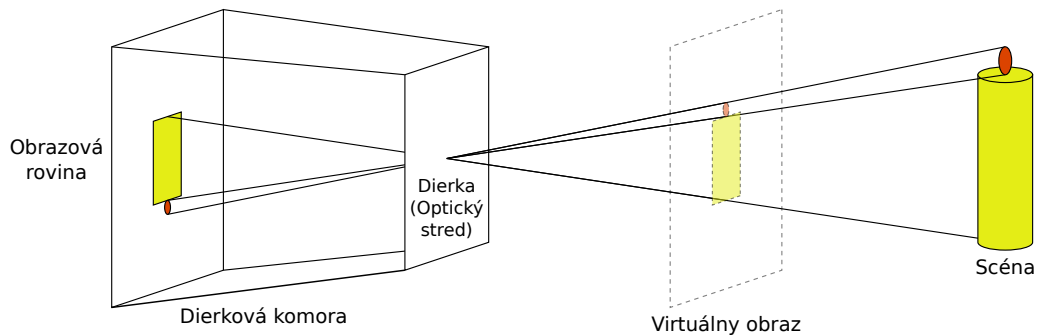
2.1 Počítačové videnie, obraz a kamera

Počítačové videnie je odbor informačných technológií zaoberajúci sa získavaním, spracovaním a analýzou informácií z obrazu. Počítačové videnie má dva ciele. Z pohľadu biológie sa počítačové videnie snaží vytvoriť výpočtové modely zrakovej sústavy človeka. Z inžinierskeho pohľadu sa snaží vytvoriť autonómne systémy, ktoré dokážu vykonávať také úlohy, aké človek vie vykonávať vďaka zrakovému zmyslu na rovnakej alebo lepšej úrovni [11]. Medzi základné úlohy patrí extrahovanie informácií o čase a priestore z obrazov nasnímaných kamerami.

Obraz môže byť definovaný ako spojitá dvojrozmerná funkcia $f(x, y)$, kde x a y sú priestorové súradnice a hodnota funkcie f v súradniciach (x, y) je intenzita alebo jas obrazu v danom bode. Existujú obrazové funkcie, ktorých hodnota určuje aj farbu, ale pre účely tejto práce stačia obrazy v odtieňoch šedej. Počítačové videnie pracuje výlučne s **digitálnymi obrazmi**. Obrazová funkcia digitálneho obrazu sa od spomínanej obrazovej funkcie líši v tom, že jej definičný obor a obor hodnôt je konečný a diskrétny. Digitálny obraz pozostáva z konečného množstva bodov, kde každý má svoju pozíciu a hodnotu. Tieto body sa nazývajú pixely [8]. Prevod spojitého obrazu na digitálny sa nazýva digitalizácia. Digitálny obraz môže byť vytvorený buď pomocou počítačových programov, alebo zachytený pomocou digitálnych senzorov. Digitálne senzory zachytávajú informácie o reálnom svete a prevádzajú ich do digitálnej podoby. Prístroje s digitálnymi senzormi, ktoré môžu vytvárať digitálne obrazy, sú napríklad fotoaparát, kamera, ale aj radar, sonar, sonograf.

Kamera je prístroj určený na zachytávanie obrazu vytvoreného svetlom. **Digitálna kamera** je kamera, ktorá zo zachyteného svetla vytvára digitálny obraz pomocou elektronického obrazového senzora a ukladá ho do digitálnej pamäte.

Keďže obraz je dvojrozmerný a realita, ktorú zachytáva, je trojrozmerná, musí v procese vytvárania obrazu dôjsť k transformovaniu trojrozmerných súradníc objektov v reálnom svete na dvojrozmerné súradnice obrazov daných objektov. Táto transformácia a princíp fungovania kamery sa dá popísať pomocou **modelu dierkovej komory** [7]. Tento model popisuje vzťah medzi bodom v trojrozmernom priestore a jeho obrazom v obrazovej rovine. Model je vyobrazený na obrázku 2.1. Pozostáva zo scény, z ktorej vychádzajú svetelné lúče, ktoré prechádzajú cez dierku (optický stred) veľkosti bodu a obrazovej roviny, na ktorú je pomocou daných lúčov daná scéna premietnutá. Takto vytvorený obraz je prevrátený. V reálnej kamere sa v pozícii obrazovej roviny nachádza obrazový senzor a namiesto diery je použitý objektív. Na zjednodušené vysvetlenie ďalších princípov v tejto kapitole je namiesto reálneho prevráteného obrazu použitý obraz virtuálny, ktorý nie je prevrátený a nachádza sa v rovnakej vzdialenosti od optického stredu, ale na opačnej strane.



Obr. 2.1: Model dierkovej komory

Matematický vzťah medzi trojrozmernými súradnicami reálneho sveta a dvojrozmernými súradnicami obrazu popisuje **kamerová projekcia**. Tá pozostáva z transformácie trojrozmerných súradníc reálneho sveta do trojrozmerných súradníc kamery a následnej perspektívnej projekcie súradníc kamery na súradnice obrazu.

Pozícia ľubovoľného bodu v reálnom svete sa dá vyjadriť pomocou trojrozmernej súradnicovej sústavy sveta. Avšak aby bolo možné popísať, ako sa daný bod zobrazí do obrazu, je potrebné poznať jeho pozíciu v rámci trojrozmernej súradnicovej sústavy kamery. Súradnicová sústava kamery má počiatok v optickom strede kamery. Jej osi x a y sú rovnobežné s obrazovou rovinou a os z je na ňu kolmá. Transformácia súradníc sveta na súradnice kamery sa dá popísať rotačnou maticou R veľkosti 3×3 a translačným vektorom T o veľkosti 3×1 . Translačný vektor je daný bodmi t a o ako $T = t - o$, kde bod o označuje optický stred kamery a t počiatok súradnicovej sústavy sveta v súradnicovej sústave kamery. Pomocou rotačnej matice sa zarovnajú smery osí súradnicovej sústavy sveta na smery osí súradnicovej sústavy kamery a pomocou translačného vektoru sa začiatok súradnicovej sústavy sveta presunie na začiatok súradnicovej sústavy kamery. Vzťah medzi bodom P_W v súradnicovom systéme sveta a jeho korešpondujúcim bodom P_C v súradnicovom systéme kamery popisuje rovnica (2.1).

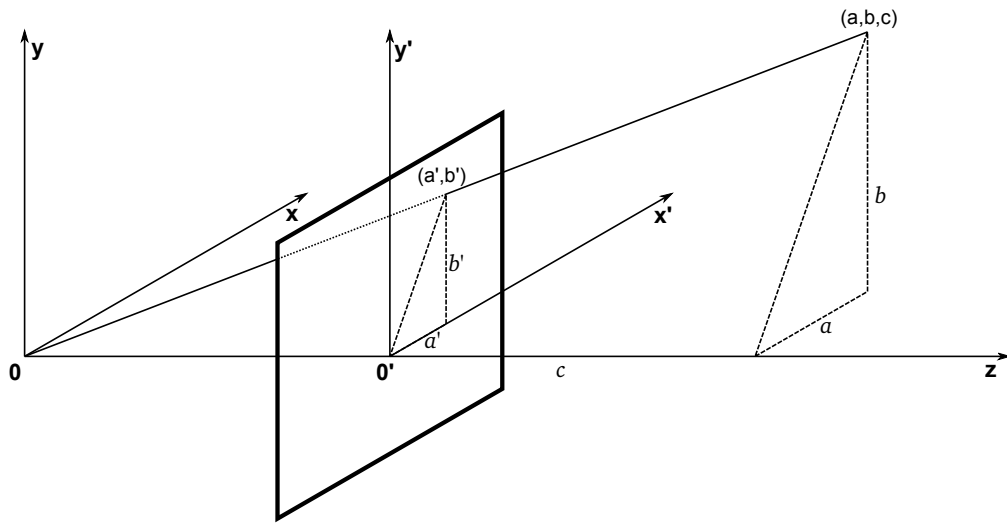
$$P_C = RP_W + T \quad (2.1)$$

Rotačná matica R a translačný vektor T sú **vonkajšie parametre kamery**. **Vnútné parametre kamery** zas popisujú perspektívnu projekciu kamery, ktorá je vysvetlená ďalej. Zisťovaním vonkajších a vnútorných parametrov kamery sa zaoberá kalibrácia. Kalibrácii kamier sa venuje podkapitola 2.3.

Keďže body v súradnicovom systéme kamery majú trojrozmerné súradnice, ale obraz je len dvojrozmerný, je potrebné vedieť, ako sa tieto body budú mapovať do obrazu. Tento proces je popísaný **perspektívnou projekciou** [1]. Model perspektívnej projekcie sa nachádza na obrázku 2.2. Model pozostáva z bodu (a, b, c) v súradnicovom systéme kamery a obrazovej roviny. Obrazová rovina je kolmá na os z súradnicového systému kamery a jej vzdialenosť od začiatku súradnicovej sústavy sa označuje ako f (ohnisková vzdialenosť). Bod (a, b, c) je premietnutý do obrazovej roviny v mieste, kde priamka vedená z daného bodu do začiatku súradnicového systému kamery pretína obrazovú rovinu. Vzťah medzi bodom (a, b, c) a jeho obrazom (a', b') sa dá vyjadriť rovnicou (2.2), alebo pomocou perspektívnej matice a homogénnych súradníc (2.3).

$$a' = f \frac{a}{c}, \quad b' = f \frac{b}{c} \quad (2.2)$$

$$\begin{bmatrix} a'' \\ b'' \\ c'' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix}, \quad a' = \frac{a''}{c''}, \quad b' = \frac{b''}{c''} \quad (2.3)$$



Obr. 2.2: Perspektívna projekcia bodu (a, b, c) súradnicového systému kamery do bodu (a', b') súradnicového systému obrazu

Obraz vytvorený perspektívnou projekciou ale ešte nie je zhodný s digitálnym obrazom, ktorý kamera vytvorí. Súradnice zachyteného obrazu (x, y) nie sú pixelové súradnice digitálneho obrazu. Súradnice obrazu bývajú väčšinou v štandardných dĺžkových jednotkách, ale súradnice digitálneho obrazu sú v pixeloch [11]. Preto je potrebný škálovací parameter, ktorý berie do úvahy veľkosť pixelov obrazového senzora. Keďže pixely nemajú vždy štvorcový tvar, ale môžu byť aj obdĺžnikové, každá os má svoj vlastný škálovací parameter. Začiatok súradnicovej sústavy digitálneho obrazu nebýva v strede, ale v rohu, takže všetky súradnice musia byť posunuté v oboch smeroch. Z tohto dôvodu existujú parametre,

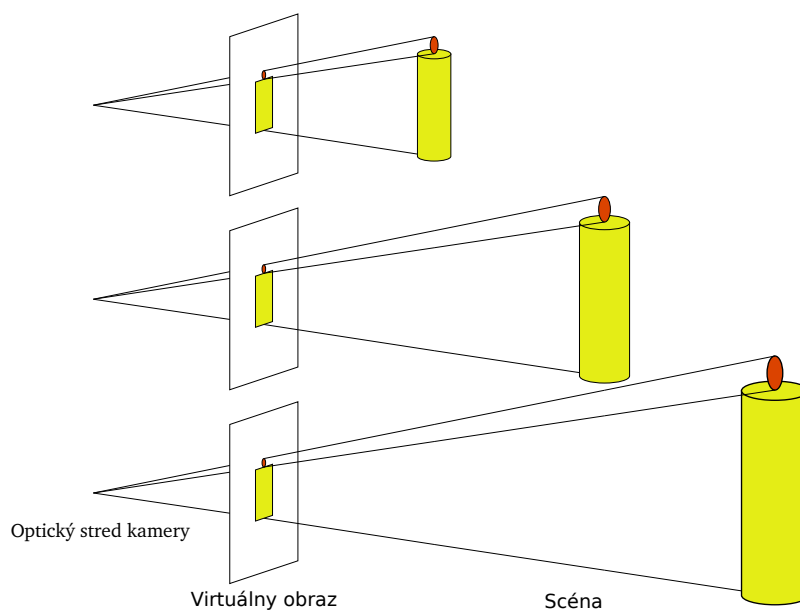
ktoré definujú posun súradnicového systému digitálneho obrazu voči obrazovej rovine v pixeloch. Rovnica popisujúca transformáciu medzi obrazom a digitálnym obrazom je potom nasledujúca:

$$p_D = \begin{bmatrix} k & 0 & x_0 \\ 0 & l & y_0 \\ 0 & 0 & 1 \end{bmatrix} p, \quad (2.4)$$

kde p_D je normalizovaný bod digitálneho obrazu, p je normalizovaný bod obrazu, k a l sú škálovacie parametre a x_0 a y_0 sú parametre posunu stredu obrazu voči digitálnemu obrazu v pixeloch. Toto sú len základné parametre, ktoré ovplyvňujú projekciu a môžu existovať ďalšie, ktoré zohľadňujú použité šošovky, ich nepresnosti, skreslenie a podobne.

2.2 Stereokamera a stereovidenie

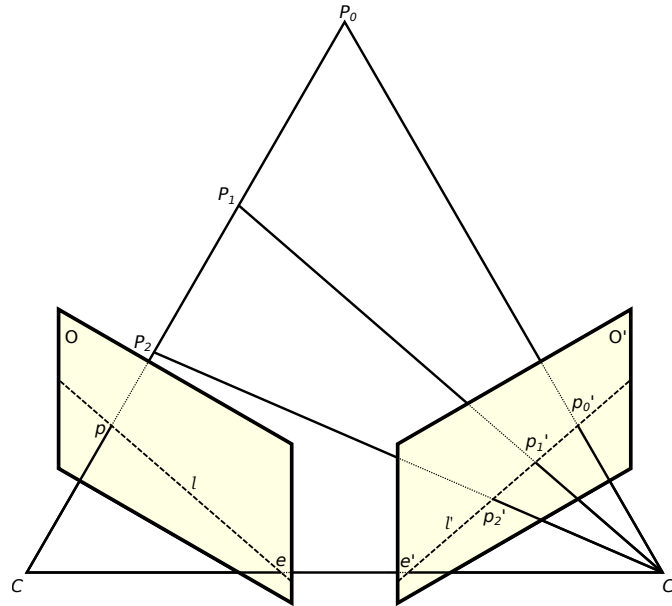
Problémom kamery je, že ak nie sú známe presné rozmery nasnímaných objektov, nedokážeme určiť ich vzdialenosť od kamery a teda ani ich presnú pozíciu v trojrozmernom priestore. Tento problém popisuje obrázok 2.3. Z obrazu nie je možné zistiť, či sa na ňom nachádza obraz malej sviečky, ktorá bola umiestnená blízko kamery alebo obraz veľkej sviečky umiestnenej ďaleko od kamery. Aby bolo možné identifikovať hĺbku scény, je potrebné pridať ďalšiu kameru, ktorá bude snímať objekty z inej pozície.



Obr. 2.3: Malý objekt umiestnený blízko kamery môže mať rovnaký obraz ako veľký objekt umiestnený ďalej od kamery

Stereokamera je prístroj pozostávajúci z dvoch synchronizovaných kamier umiestnených zvyčajne horizontálne vedľa seba na spoločnej základni. Stereokamera sníma dve snímky (pravú a ľavú) súčasne. Stereovidenie sa zaoberá získavaním informácií práve zo snímok pochádzajúcich zo stereokamery. Jedným z problémov, ktoré musí stereovidenie riešiť, je zistiť, kde sa objekt zobrazený v jednom obraze nachádza v druhom obraze. Dvojica bodov, jeden z prvého obrazu a druhý z druhého, ktoré zobrazujú ten istý bod v priestore,

sa nazývajú korešpondujúce body. Základné metódy na hľadanie korešpondujúcich bodov sú popísané v kapitole 3. Hľadanie korešpondujúcich bodov na celých snímkach by bolo časovo veľmi náročné, preto sa pri stereokamere využíva **epipolárna geometria**.



Obr. 2.4: Epipolárna geometria dvoch kamier: Kamery sú reprezentované svojimi optickými stredmi C , C' a svojimi virtuálnymi obrazmi O a O' . Body P_0 , P_1 , P_2 sú viditeľné body v priestore umiestnené na priamke CP_0 , p je ich spoločný obraz v ľavej kamere a p'_0 , p'_1 , p'_2 sú ich obrazy v pravej kamere v tomto poradí.

Epipolárna geometria popisuje, kde sa môže zobraziť obraz bodu v druhej kamere, keď poznáme jeho pozíciu v obraze prvej kamery. Epipolárna geometria je vyobrazená na obrázku 2.4. C a C' sú optické stredy ľavej a pravej kamery a O a O' sú ich obrazové roviny v tomto poradí. Body P_0 , P_1 a P_2 sú body v priestore ležiace na priamke CP_0 , ktorá pretína obrazovú rovinu O v bode p . Preto body P_0 , P_1 a P_2 majú spoločný obraz v pravej kamere a to bod p . Vektory CP_0 a $C'P_0$ tvoria **epipolárnu rovinu** [11]. Táto rovina pretína obrazovú rovinu ľavej kamery O v priamke l a obrazovú rovinu pravej kamery v priamke l' . Priamky l a l' sa nazývajú **epipoláry**. Epipoláry prechádzajú **epipólmi**. Epipól je priesečník obrazu s priamkou spájajúcou optické stredy kamier, respektíve epipól je obraz optického stredu jednej kamery v obraze druhej kamery. Epipolára l prechádza epipólom e a epipolára l' prechádza epipólom e' . Všetky body v tejto epipolárnej rovine sa môžu zobraziť do ľavej kamery len ako body na epipoláre l a do pravej kamery len ako body na epipoláre l' . Ak by bod p predstavoval jeden bod v priestore, tak nie je možné povedať, či sa jedná o bod P_0 , P_1 , P_2 alebo o úplne iný bod na polpriamke Cp , a teda nie je možné povedať, či obraz tohto bodu v pravej kamere bude v bode p'_0 , p'_1 , p'_2 alebo iný, ale je možné povedať, že ten obraz sa bude nachádzať na epipoláre l' . Toto sa nazýva **epipolárne obmedzenie**. Vďaka epipolárnemu obmedzeniu stačí pri hľadaní korešpondenčných bodov prehľadávať body na epipoláre a nie je potrebné prehľadávať celý obraz.

Aby bolo možné zistiť, kde sa nachádza epipolára bodu, je potrebné poznať vzájomné postavenie kamier stereokamery. Takisto ako vzťah medzi súradnicovým systémom sveta a súradnicovým systémom kamery, ktorý je popísaný translačným vektorom a rotačnou

maticou v podkapitole 2.1, tak aj vzťah medzi ľavou a pravou kamerou stereokamery sa dá popísať rovnakým spôsobom. Ak translačný vektor T je vektor z optického stredu ľavej kamery C do optického stredu pravej kamery v súradnicovom systéme ľavej kamery C' a rotačná matica R popisuje rotáciu medzi súradnicovým systémom ľavej a pravej kamery, tak platí rovnica [11]:

$$\mathbf{p}^T \mathcal{E} \mathbf{p}' = 0, \quad (2.5)$$

kde \mathbf{p} a \mathbf{p}' sú homogénne vektory z C do p a p' v tomto poradí a $\mathcal{E} = [T]_{\times} R$, kde $[T]_{\times}$ je maticová reprezentácia vektorového súčinu vektoru T , kde platí $[T]_{\times} x = T \times x$ pre ľubovoľný vektor x . Matica \mathcal{E} sa nazýva esenciálna (essential matrix) a popisuje vzťah medzi bodom a jeho epipolárom:

$$\mathbf{p}^T \mathcal{E} = \mathbf{l}', \quad (2.6)$$

kde \mathbf{l}' je vektorová forma priamky l' . Pre vzťah medzi bodom z druhého obrazu a jeho epipolárom platí:

$$\mathbf{p}'^T \mathcal{E}^T = \mathbf{l}. \quad (2.7)$$

Ak je známa esenciálna matica, tak je jednoduché vypočítať, kde v pravom obraze sa nachádza epipolára bodu v ľavom obraze a naopak.

Väčšinou sú však známe súradnice bodu v digitálnom obraze jednej kamery a je potrebné vypočítať pozíciu korešpondujúcej epipoláry v digitálnom obraze druhej kamery. Na to je možné použiť fundamentálnu maticu (fundamental matrix) [11]. Fundamentálnu maticu je možné vypočítať z esenciálnej matice a matíc vnútorných parametrov oboch kamier (matica v rovnici (2.4)):

$$\mathcal{F} = \mathcal{K}^{-T} \mathcal{E} \mathcal{K}'^{-1}, \quad (2.8)$$

kde \mathcal{K} a \mathcal{K}' sú matice vnútorných parametrov ľavej a pravej kamery. Pre epipoláry potom platia podobné rovnice

$$\mathbf{p}_D^T \mathcal{F} = \mathbf{l}'_D \quad (2.9)$$

a

$$\mathbf{p}_D'^T \mathcal{F}^T = \mathbf{l}_D, \quad (2.10)$$

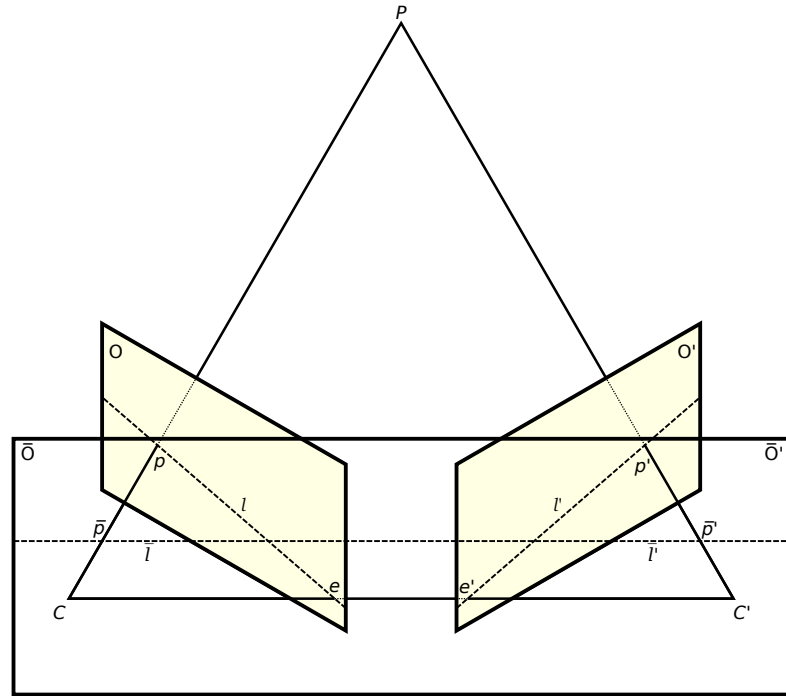
ale už v súradniciach digitálnych obrazov.

Proces hľadania korešpondujúcich bodov v stereosnímke sa dá ešte ďalej zjednodušiť, a to tak, že všetky korešpondujúce epipoláry budú zhodné a rovnobežné s priamkou pretínajúcou optické stredu kamier. Takýto proces úpravy epipolár sa nazýva rektifikácia [19]. Rektifikácia nahrádza obrazy stereokamery dvomi ekvivalentnými obrazmi, ktoré už však majú spoločnú obrazovú rovinu. Toto je možné vidieť na obrázku 2.5. Takto rektifikované obrazy stačí prehľadávať pre korešpondujúce body už len v horizontálnom smere, čo celý proces zjednodušuje a urýchľuje. Korešpondujúci bod sa totiž nachádza na rovnakej vertikálnej súradnici.

2.3 Existujúce kalibračné postupy a ich presnosť

Kalibrácia kamery je proces zisťovania vnútorných (intrinsic parameters) a vonkajších (extrinsic parameters) parametrov kamery. Pri kalibrácii stereokamery je potrebné spočítať esenciálnu a fundamentálnu maticu. Kalibrácia je neoddeliteľnou súčasťou počítačového stereovidenia, ktorá je potrebná na správne identifikovanie pozície sledovaných objektov.

Vnútorné parametre kamery popisujú vzťah medzi súradnicovým systémom danej kamery a súradnicovým systémom modelu dierkovej komory [11]. Pozostávajú z interných



Obr. 2.5: Rektifikácia: Obrazy O a O' sú premietnuté do obrazov \bar{O} a \bar{O}' , ktoré majú spoločnú obrazovú rovinu paralelnú na priamku prechádzajúcu optickými stredmi kamier CC' . Epipoláry l a l' sú premietnuté do jednej priamky $\bar{l} = \bar{l}'$ rovnobežnej s CC' . Body p a p' sú premietnuté do bodov \bar{p} a \bar{p}' , ktoré sa nachádzajú na priamke $\bar{l} = \bar{l}'$.

geometrických a optických vlastností kamery ako ohnisková vzdialenosť, poloha geometrického stredu a skreslenie [26].

Vonkajšie parametre kamery popisujú pozíciu a orientáciu kamery voči priestoru, v ktorom sa nachádza. Vonkajšie parametre sú obzvlášť dôležité pri stereokamere. V tomto prípade sú obrázky snímané z dvoch rôznych pozícií. Vonkajšie parametre popisujú vzájomnú pozíciu a orientáciu medzi ľavou a pravou kamerou stereokamery, vďaka čomu je možné určovať hĺbku nasnímanej scény. Presnosť kalibračných postupov je veľmi dôležitá, lebo čím presnejšie je stereokamera skalibrovaná, tým presnejšie je možné merať pozíciu, pohyb a rýchlosť sledovaných objektov.

Kalibrácia kamery pozostáva z dvoch častí: nasnímania objektu alebo scény a následného vypočítania kalibračných parametrov pomocou informácií získaných z nasnímaných snímok. Podľa typu objektu a scény sa kalibračné metódy rozdeľujú na kontrolované (supervised) a nekontrolované (unsupervised) [28]. Pri kontrolovaných metódach sa sníma objekt, ktorého presné parametre ako veľkosť a tvar sú dopredu známe. Nekontrolované metódy takéto objekty nepotrebnú.

Jedna z často používaných kontrolovaných metód kalibrácie je kalibrácia pomocou šachovnicového vzoru. Túto metódu popísal Z. Zhang vo svojej práci [30]. Ako kalibračný objekt sa používa plochý (dvojmerný) vzor, ktorý je možné vytlačiť na tlačiarňu. Rozmery vzoru musia byť známe. Najčastejšie sa používa šachovnicový vzor, lebo obsahuje vysoko kontrastné rohy, ktoré je jednoduché detekovať v nasnímanom obraze. Vzor stačí nasnímať v dvoch rôznych orientáciách, pohyb objektu alebo kamery nemusí byť známy. Pre presnejšie výsledky sa ale používajú viac ako dve snímky. Metódu Z. Zhanga napríklad

využili Jalalat *et al.* v ich práci [12] a Yang *et al.* v ich práci [29]. V obidvoch prácach skalibrovali stereokameru pomocou šachovnicového vzoru, ktorou potom snímali vozidlá v premávke a na základe spracovania nasnímaných snímok zisťovali polohu a rýchlosť zachytených vozidiel.

Existujú aj plne automatické spôsoby kalibrácie. Jednou z nich je metóda od Sochor *et al.* popísaná v ich práci [24]. Táto metóda je určená na sledovanie cestnej premávky a počítanie rýchlosti prechádzajúcich vozidiel pomocou jednej monokulárnej kamery. Problémom plne automatických metód je, že potrebujú odhadnúť mierku. Aby bolo možné zmerať rýchlosť vozidla, je potrebné poznať vzdialenosť, akú vozidlo prešlo a za aký čas to stihlo. Na určenie vzdialenosti je potrebné poznať presnú veľkosť objektu v scéne. Daná metóda používa neurónovú sieť na rozpoznanie a presné zaradenie prechádzajúcich vozidiel. Z presných rozmerov zaradených áut je potom vypočítaná mierka. Na zisťovanie ostatných parametrov kalibrácie sú použité dva úbežné body. Úbežný bod je bod v obrazovej rovine, v ktorom sa zdá, že rovnobežné čiary v trojrozmernom priestore zobrazené v obrazovej rovine konvergujú.

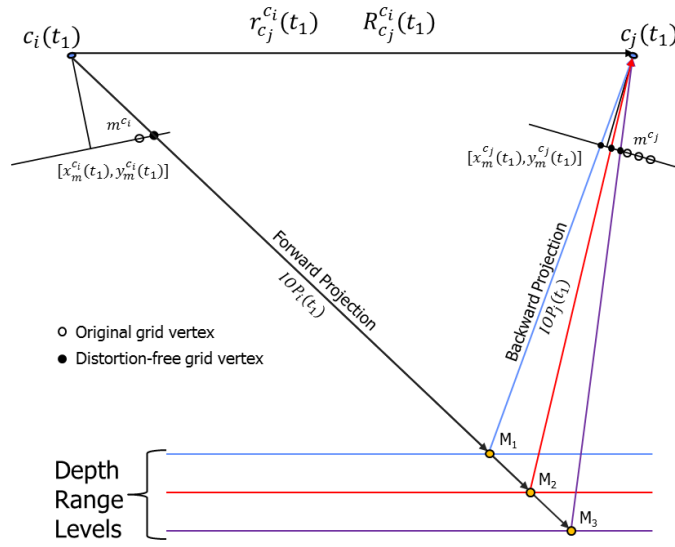
Okrem spomenutých postupov existuje aj mnoho ďalších, ktoré sa medzi sebou líšia presnosťou, rýchlosťou a mierou automatizácie.

2.4 Existujúce práce zaoberajúce sa presnosťou stereokamier

Dopravné stereokamery sú zvyčajne určené na meranie dráhy a rýchlosti prechádzajúcich vozidiel. Problém nastáva, ak sa skalibrovaná kamera pohne voči priestoru, v ktorom sa nachádza, a v prípade stereokamery aj keď sa jedna kamera pohne voči druhej. Získané informácie o scéne snímanej kamerou, ktorá sa počas snímania pohne, potom môžu byť chybné. Posun pri dopravných kamerách môže byť krátkodobý (napríklad vibrácie prechádzajúcich ťažkých áut môžu rozpochybovať kameru), alebo dlhodobý (spôsobený napríklad rozťažnosťou materiálov vplyvom tepla). V preskúmaných prácach zaoberajúcich sa dopravnou stereokamerou sa buď nespomína spôsob, akým je zabezpečené, aby sa kamera nepohybovala [12], alebo je kamera umiestnená na statíve alebo podobnom zariadení a pohyby kamier sa v meraniach ignorujú [5][6][29].

Habib *et al.* sa v ich práci [9] zaoberajú analýzou stability multikamerového systému. Pomocou troch rôznych metód sa snažia zistiť, aký vplyv má (ne)stabilita interných a externých parametrov kamier na 3D rekonštrukciu. Všetky tri navrhované metódy sú založené na rovnakom princípe. V obraze prvej kamery je definovaná virtuálna mriežka obrazových bodov. Podľa interných a externých parametrov z prvej kalibrácie tejto kamery je spočítaná pozícia v súradnicovej sústave sveta týchto bodov vo viacerých vzdialenostiach od kamery. Ďalej sú podľa interných a externých parametrov z prvej kalibrácie druhej kamery spočítané súradnice mriežok v tejto kamere, obrázok 2.6. Následne je pomocou jednej z troch metód odhadnutá chyba v obrazových jednotkách, ktorá by vznikla, ak by boli použité kalibračné parametre z kalibrácie vykonanej v inom čase. Prvá metóda využíva na výpočet nových súradníc doprednú a spätnú projekciu. Druhá metóda je zameraná na spočítanie dopredných projekcií mriežky z obidvoch kamier a spočítanie vzdialenosti medzi korešpondujúcimi bodmi mriežok v obrazových jednotkách. Tretia metóda počíta rozdiely v spoločnom normalizovanom obraze obidvoch kamier. Na otestovanie bol zostavený systém pozostávajúci zo siedmich kamier pripevnených na kovovom ráme. Každá kamera systému bola trikrát skalibrovaná podľa plochého vzoru. Následne boli spočítané jednotlivé chyby každou z troch metód medzi každou možnou dvojicou kamier v každej kombinácii kalibrácií. Rozsah spo-

čítaných chýb bol 0.21 až 2.04 pixelu s priemerom 0.82 pre prvú metódu. Druhá metóda mala rozsah chýb 0.12 až 0.56 s priemerom 0.3 pixelu a tretia metóda mala rozsah chýb 0.08 až 1.24 s priemerom 0.63. Následne bola položená hypotéza, že systém je stabilný ak sa chyba pohybuje v ráde jedného pixelu. Ak je systém stabilný, výsledky rekonštrukcie budú kompatibilné bez ohľadu na to, ktorá z kalibrácií je použitá. Táto hypotéza bola overená na 3D rekonštrukcii figuríny pomocou všetkých troch kalibrácií. Hypotéza bola potvrdená, keďže žiadna z kalibrácií nepriniesla výrazne odlišné výsledky.



Obr. 2.6: Poloha mriežky v súradniciach obrazu prvej kamery a jej projekcia do svetových súradníc vo viacerých hĺbkach a ich spätná projekcia do obrazu druhej kamery [9].

Pan *et al.* sa v ich práci [20] zaoberajú vplyvom tepelných zmien na presnosť Stereo-DIC (Stereo digital image correlation – korelácia digitálnych stereosnímkov). Na testovanie použili stereokameru sledujúcu vzor na optickom kremennom skle. Toto sklo bolo vybrané preto, lebo jeho koeficient tepelnej expanzie je blízky nule. Počas experimentovania bola snímaná teplota skla a obidvoch kamier. Boli prevedené dva testy. Prvý mal simulovať použitie kamery vo vnútorných priestoroch, preto bola počas neho udržiavaná konštantná teplota okolia. Druhý test mal simulovať podmienky vonku, preto bola menená teplota okolia pomocou klimatizácie. Pred obidvomi testami boli kamery vypnuté a mali rovnakú teplotu ako okolie. Na začiatku každého testu bolo nasnímaných 30 stereosnímkov vzoru z rôznych pozícií. Tie boli použité na skalibrovanie kamier v pôvodnom stave. Potom boli kamery ponechané, aby 5 hodín snímali vzor na skle. Z tohto snímania bola každé dve minúty uložená jedna stereosnímkov. Na koniec prvého testu bolo znovu zaznamenaných 30 stereosnímkov vzoru z rôznych pozícií, ktoré boli použité na zistenie parametrov stereokamery po snímaní, keď kamera bola ešte zohriata zo snímania. Zaznamenané snímky boli použité na výpočet polí trojrozmerných posunutí (3D displacement fields) snímaného vzoru. Výsledky ukázali, že tie posunutia boli v korelácii so zaznamenanými zmenami teploty kamier.

Chu sa vo svojej práci [2] zaoberá stabilizáciou stereovidea vytvoreného mobilným telefónom. Videá natočené pomocou mobilných telefónov sú často nestabilné a čo je spôsobené

malými rozmermi a váhou mobilných telefónov a neschopnosťou užívateľa držať telefón stabilne. Autor článku navrhol a naimplementoval program, ktorým je možné z takéhoto videa odstrániť vysokofrekvenčné pohyby, ktoré sú rušivé pre divákov. Použitý algoritmus pozostáva z troch častí. V prvej časti algoritmu sú nájdené korešpondujúce body medzi dvomi snímkami stereosnímky. V druhej časti počíta pohyb kamery. Na tento výpočet využíva informácie z gyroskopu a akcelerometra, ktoré sú súčasťou mobilného telefónu. Pohyb kamery je modelovaný 3D rotačnou maticou a 3D translačnou maticou. V treťom kroku sú zo získaného pohybu odstránené vysokofrekvenčné pohyby. Následne sú z pôvodných snímkov syntetizované snímky, ktoré sa nachádzajú už na stabilizovanej trase kamery. Na to je použitý algoritmus DIBR (Depth Image Based Redering – renderovanie snímkov za pomoci hĺbkovej mapy). Nakoniec je ešte odstránená vertikálna odchýlka medzi snímkami stereosnímky, ktorá mohla vzniknúť pri renderovaní.

Kapitola 3

Algoritmy spracovania obrazu

Táto kapitola sa zaoberá algoritmami, ktoré sa používajú na hľadanie transformácií medzi dvoma obrazmi snímajúcimi rovnakú scénu z rôznych miest. Nejedná sa o encyklopedický prehľad týchto algoritmov, tých existuje príliš veľké množstvo na to, aby boli všetky popísané v tejto práci. Boli vybrané metódy so zameraním na stereospracovanie obrazu.

Aby bolo možné vypočítať transformáciu medzi dvoma obrazmi, je potrebné nájsť objekty, ktoré sa nachádzajú na oboch obrazoch. Čo a akým spôsobom sa najčastejšie hľadá v obrazoch, je popísané v prvej podkapitole. Druhá podkapitola sa zaoberá spôsobmi, ktorými sa určuje, či nájdené objekty sú zhodné. Tretia podkapitola sa venuje transformáciám a ich výpočtu. Štvrtá podkapitola sa venuje procesu párovania bodov z rôznych obrazov. Posledná podkapitola sa venuje prevzorkovaniu a prečo je potrebné pri aplikovaní transformácií na obrazy.

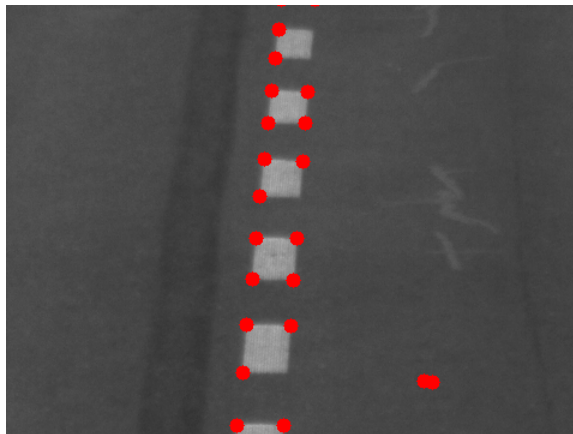
3.1 Hľadanie vhodných bodov na sledovanie

Hľadanie vhodných bodov na sledovanie je podobné procesu skladania puzzle [18]. Kúsok puzzle s jednoliatou textúrou je veľmi ťažké zaradiť na miesto kam patrí. Ak sa však na danom kúsku nachádza hrana objektu, existuje podstatne menšie množstvo pozícií, kde sa má kúsok nachádzať. Ideálne je, keď sa na kúsku nachádza roh. Spravidla potom ostáva už len minimálny počet miest, na ktoré by daný kúsok mohol pasovať. Na obrázku 3.1 je možné vidieť rohy nájdené pomocou jedného z detektorov rohov popísaných nižšie.

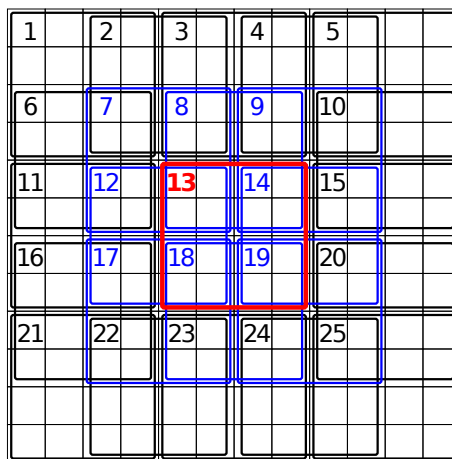
Asi najjednoduchším detektorom rohov je **Moravcov detektor** [15]. Moravcov detektor funguje na princípe hľadania maxima rozdielu jasů príslušných oblastí. Jednoliaté textúry majú vo všetkých smeroch rozdiel jasů minimálny a hrany taktiež nemajú veľký rozdiel jasů v smere hrany. Rohy ako jediné majú rozdiel jasů vo všetkých smeroch. Moravcov detektor rozdelí obraz na prekrývajúce sa okná, ako na obrázku 3.2. Pre každé okno je vypočítaný jeho rozdiel jasů. Pre každé okno je spočítaná suma rozdielov druhých mocnín jasů v štyroch smeroch:

- horizontálne: $\sum(I_{i,j} - I_{i,j+1})^2$,
- vertikálne: $\sum(I_{i,j} - I_{i+1,j})^2$,
- diagonálne v oboch smeroch: $\sum(I_{i,j} - I_{i+1,j+1})^2$ a $\sum(I_{i,j} - I_{i+1,j-1})^2$.

$I_{x,y}$ je intenzita jasů pixelu na súradniciach (x, y) . Minimálna hodnota z týchto štyroch súm je potom rozdiel jasů okna. Okno je označené ako okno s rohom, ak jeho rozdiel jasů je lokálne maximum 25 príslušných a prekrývajúcich sa okien.



Obr. 3.1: Rohy (červené body) nájdené na obrázku pomocou Shi-Tomasi detektora rohov



Obr. 3.2: 8 prekrývajúcich sa okien(modré) a 16 priliehajúcich okien(čierne) k aktuálnemu oknu (červené). Najmenšie bunky predstavujú pixely. Každé okno má veľkosť 4x4 pixely. Okno je prehlásené za roh len v prípade, že jeho rozdiel jasů je maximum spomedzi týchto 25 okien.

Tento prístup má svoje chyby. Jeden z jeho nedostatkov je, že detektor označí ako rohy aj hrany, ktoré nie sú orientované v jednom zo štyroch smerov, pre ktoré detektor počíta sumy štvorcov rozdielov.

Z Moravcovho detektora vychádza **Harrisov detektor** [27]. Namiesto počítania rozdielu jasů len v štyroch smeroch, ho počíta všeobecne vo všetkých smeroch podľa rovnice (3.1), kde $w(x, y)$ je oknová funkcia, $I(x, y)$ je jas vzorky a $I(x + u, y + v)$ je jas vzorky posunutej o vektor (u, v) .

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (3.1)$$

Táto funkcia dosahuje svoje maximum práve v rohoch. Rovnica sa dá upraviť pomocou Taylorovho rozkladu na rovnicu [17]:

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.2)$$

kde

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \quad (3.3)$$

I_x a I_y sú parciálne derivácie jasu obrazu v x-ovej a y-ovej ose. Stačí vypočítať rovnicu

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(M) - k(\text{trace}(M))^2 \quad (3.4)$$

kde

- λ_1 a λ_2 sú vlastné čísla matice
- $\det(M)$ je determinant matice M
- $\text{trace}(M)$ je súčet čísel na diagonále matice M (stopa matice)
- k je parameter, ktorého hodnoty z intervalu $\langle 0.04; 0.15 \rangle$ boli empiricky vyhodnotené ako prípustné

Podľa hodnoty R (kritérium určovania rohov) sa dá potom určiť, či sa jedná o roh alebo nie:

- Ak je $|R|$ malé, tak aj λ_1 a λ_2 sú malé a daná oblasť je plochá bez rohov a hrán
- Ak je R záporné, tak $\lambda_1 \gg \lambda_2$ alebo $\lambda_1 \ll \lambda_2$ a jedná sa o hranu
- Ak je R veľké, tak aj λ_1 a λ_2 sú veľké a približne rovnaké a v oblasti sa nachádza roh

Veľmi podobný Harrisovmu detektoru je detektor **Shi-Tomasi** [23]. Jediný rozdiel je v počítaní hodnoty R , Shi-Tomasi ju počíta nasledujúcim spôsobom:

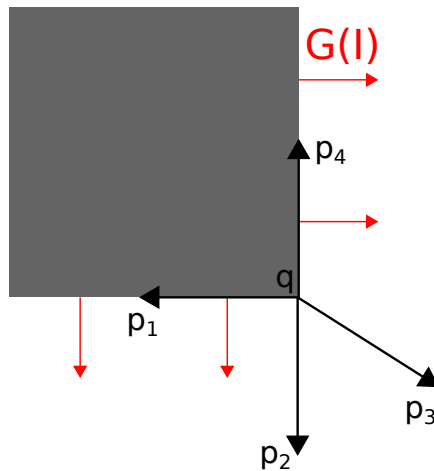
$$R = \min(\lambda_1, \lambda_2). \quad (3.5)$$

Tento detektor je pomalší, keďže priamo musí počítať vlastné čísla matice. Tento algoritmus je však podľa jeho tvorcov lepší ako Harrisov detektor.

Spomenuté detektory nie sú zďaleka jediné metódy na hľadanie rohov v obraze. Existuje rada ďalších metód, ktoré sa medzi sebou líšia v náročnosti na výpočet a v kvalite.

Väčšina metód na hľadanie rohov používa presnosť na pixelovej úrovni. Niekedy však táto presnosť môže byť nedostačujúca. Preto existujú metódy na hľadanie rohov na **sub-pixelovej úrovni**. Väčšinou využívajú ako základ na vyhľadanie rohov vyššie spomenuté metódy a pozíciu rohov už len upresňujú. Jeden z postupov vychádza z teórie ortogonálnych vektorov [31]. Každý vektor z rohu q do bodu p_i nachádzajúceho sa v blízkosti rohu, by mal byť kolmý na gradient obrazu v bode p_i , obrázok 3.3. V ideálnom prípade by súčin týchto dvoch navzájom kolmých vektorov bol nula, ale šum v obraze spôsobuje chybu ε_i :

$$\varepsilon_i = G(I_{p_i})^T (q - p_i) \quad (3.6)$$



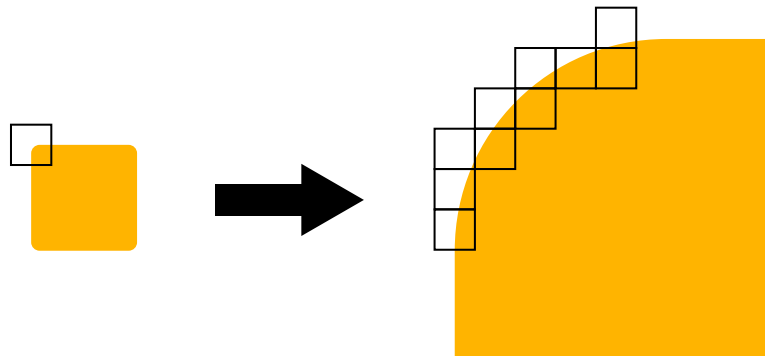
Obr. 3.3: Vektory (q, p_i) (čierna) kolmé na gradient $G(I)$ (červená)

$G(I_{p_i})$ je gradient obrazu v bode p_i . Po vynásobení oboch strán rovnice gradientom $G(I_{p_i})$ a predpokladaním, že chyba ε_i je nula, získavame sústavu rovníc:

$$0 = \sum_i (G(I_{p_i})G(I_{p_i})^T)q - \sum_i (G(I_{p_i})G(I_{p_i})^T p_i) \quad (3.7)$$

Po vypočítaní q bude nový bod q považovaný za roh. Tento postup sa môže opakovať, kým nebude dosiahnutá požadovaná presnosť rohu alebo nebude prekročený maximálny počet iterácií.

Rohy sa dajú použiť na hľadanie rovnakých objektov v rôznych obrázkoch, avšak nie sú vždy ideálne. V prípade rôznej mierky obrázkov to čo bolo považované za roh objektu v obrázku, kde bol objekt malý, už nemusí byť považované za roh na tom istom priblíženom objekte. Tento fenomén vysvetľuje obrázok 3.4.



Obr. 3.4: Na obrázku vľavo sa na žltom objekte nachádza jasný roh, ale na obrázku vpravo na tom istom priblíženom objekte už je možné vidieť len mierne zakrivenú hranu

Riešením nielen tohto problému je nájsť na obrázku informácie (features) o objektoch, ktoré sú nemenné vzhľadom na posun, rotáciu, zmenu mierky a zmenu osvetlenia objektu. Taktiež je potrebné vedieť určiť, či informácia extrahovaná z jedného obrázka je zhodná s informáciou z iného obrázka.

Jedným z algoritmov, ktoré sa snažia tento problém riešiť, je algoritmus Scale Invariant Feature Transform (Transformácia invariantných vlastností vzhľadom na zmenu mierky) [14]. Tento algoritmus je tu popísaný len zjednodušene. Hľadanie informácií z obrázkov prebieha v štyroch častiach. V prvej časti sa vyhľadávajú kľúčové body, ktoré sú nemenné v rôznych mierkach obrázku. V druhej časti sú kľúčové body prefiltrované. Odstránené sú tie kľúčové body, ktoré majú nízky kontrast a tie, ktoré sa nachádzajú na hranách. Pozície ostávajúcich kľúčových bodov sú upresnené. V tretej časti sú kľúčovým bodom priradené orientácie. Orientácia kľúčového bodu je určená podľa smeru, v ktorom sa nachádza maximálny gradient v okolí kľúčového bodu. V štvrtej časti je každý kľúčový bod popísaný 128 hodnotami. Tieto hodnoty sa využívajú pri párovaní kľúčových bodov z iných obrázkov.

3.2 Zisťovanie podobnosti rôznych bodov

Keď sú už nájdené rohové body vo viacerých obrázkoch, je potrebné zistiť, ktoré body z jedného obrázku sú zhodné s bodmi z ďalšieho obrázku. Proces zisťovania podobnosti bodov je založený na princípe porovnávania okolí jednotlivých bodov. Na to sa používajú výrezy rovnakej veľkosti so stredom v porovnávaných bodoch. Na porovnanie výrezov existuje niekoľko spôsobov.

Jedným zo spôsobov výpočtu podobnosti výrezov je **suma absolútnych rozdielov** [21]. Od každého jasu pixelu v prvom výreze je odčítaný jas korešpondujúceho pixelu v druhom výreze. Suma všetkých rozdielov v absolútnej hodnote potom dáva absolútny rozdiel dvoch výrezov, rovnica (3.8). Čím je rozdiel väčší, tým je podobnosť výrezov menšia. Rozdiel zhodných výrezov je nula.

$$R(I_1, I_2) = \sum_{x,y} |I_1(x, y) - I_2(x, y)| \quad (3.8)$$

Veľmi podobná metóda je **suma rozdielov štvorcov** [10]. Namiesto absolútnej hodnoty rozdielov je spočítaná druhá mocnina rozdielov, rovnica (3.9). Na rozdiel od sumy absolútnych rozdielov, kde podobnosť klesala lineárne so zväčšovaním rozdielu medzi výrezmi, pri tejto metóde klesá kvadraticky.

$$R(I_1, I_2) = \sum_{x,y} (I_1(x, y) - I_2(x, y))^2 \quad (3.9)$$

Ďalší spôsob určenia podobnosti je **vzájomná korelácia** [3]. Jedná sa o sumu súčinov jasu korešpondujúcich pixelov, rovnica (3.10). Problém tejto metódy je však ten, že čím vyšší je jas výrezov, tým vyššia je ich vzájomná korelácia, bez ohľadu na ich reálnu podobnosť. Tento problém je možné vyriešiť odčítaním priemernej hodnoty jasu jedného z výrezov od hodnôt jasu všetkých jeho pixelov, rovnica (3.11). \bar{I} je priemerná hodnota jasu výrezu.

$$R(I_1, I_2) = \sum_{x,y} (I_1(x, y) \cdot I_2(x, y)) \quad (3.10)$$

$$R(I_1, I_2) = \sum_{x,y} ((I_1(x, y) - \bar{I}_1) \cdot I_2(x, y)) \quad (3.11)$$

Predstavené metódy si neporadia s rôznym osvetlením scény. Preto vznikli **normalizované verzie** sumy rozdielov štvorcov a vzájomnej korelácie. Pre každý pixel sa vypočíta

nová normalizovaná hodnota jas. Od pôvodnej hodnoty je odčítaná priemerná hodnota jas výrezu a následne je výsledná hodnota vydelená štandardnou odchýlkou jas výrezu [3]:

$$\hat{I}(x, y) = \frac{I(x, y) - \bar{I}}{\sqrt{\sum_{x,y} (I(x, y) - \bar{I})^2}} \quad (3.12)$$

Normalizovaná verzia vzájomnej korelácie má oproti jej nenormalizovanej verzii ešte jednu výhodu. Výsledok normalizovanej korelácie musí spadať do intervalu $\langle -1, 1 \rangle$ kde 1 značí úplnú zhodu.

3.3 Transformácie obrazu

Vzťah medzi dvomi obrazmi scény popisuje transformácia. **Homografia** je transformácia mapujúca body z jedného obrazu na korešpondujúce body v druhom obraze nachádzajúce sa v jednej projektívnej rovine. Korešpondujúce body $p \leftrightarrow p'$ sú vyjadrené pomocou homogénnych súradníc: $p = (p_x, p_y, 1)^T$ a $p' = (p'_x, p'_y, 1)^T$. Homografia je vyjadrená pomocou matice H :

$$p = Hp' \Leftrightarrow \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \\ 1 \end{bmatrix} \quad (3.13)$$

H je homografická matica, ktorá má 8 stupňov voľnosti [4].

Okrem homografie existujú aj ďalšie typy transformácií, ktoré sa používajú v počítačovej grafike využívajúce maticu o veľkosti 3×3 . Jednou z nich je **afinná transformácia**. Afinná transformácia má 6 stupňov voľnosti:

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

Afinnou transformáciou je možné vyjadriť rotáciu, transláciu, zmeny veľkosti, skosenie a ich kombinácie:

- **Rotácia** o uhol θ :

$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

- **Translácia** (posunutie) o vektor $u = (u_x, u_y)$:

$$T = \begin{bmatrix} 1 & 0 & u_x \\ 0 & 1 & u_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

- **Zmena veľkosti** vektorom $v = (v_x, v_y)$:

$$\begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

- **Skosenie** faktorom skosenia $w = (w_x, w_y)$:

$$\begin{bmatrix} 1 & w_x & 0 \\ w_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Na vytvorenie zloženej afinnej transformácie stačí matice týchto elementárnych transformácií vynásobiť:

$$RT = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & u_x \\ 0 & 1 & u_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & u_x \cos \theta - u_y \sin \theta \\ \sin \theta & \cos \theta & u_x \sin \theta + u_y \cos \theta \\ 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

Pri skladaní transformácií záleží na poradí:

$$RT = \begin{bmatrix} \cos \theta & -\sin \theta & u_x \cos \theta - u_y \sin \theta \\ \sin \theta & \cos \theta & u_x \sin \theta + u_y \cos \theta \\ 0 & 0 & 1 \end{bmatrix} \neq \begin{bmatrix} \cos \theta & -\sin \theta & u_x \\ \sin \theta & \cos \theta & u_y \\ 0 & 0 & 1 \end{bmatrix} = TR \quad (3.20)$$

Afinnú transformáciu je možné alternatívne zapísať aj rovnicou (3.21). A je transformačná matica a podmatica matice (3.14), \vec{t} je translačný vektor extrahovaný z matice (3.14): $\vec{t} = (a_3, a_6)^T = (t_x, t_y)^T$. Korešpondujúce body $p \leftrightarrow p'$ sú vyjadrené pomocou kartézskych súradníc [25].

$$p = Ap' + \vec{t} \Leftrightarrow \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \Leftrightarrow \begin{cases} p_x = a_1 p'_x + a_2 p'_y + t_x \\ p_y = a_4 p'_x + a_5 p'_y + t_y \end{cases} \quad (3.21)$$

Ak je známa transformačná matica, je jednoduché z pôvodného obrazu vypočítať jeho transformovanú podobu. Niekedy je ale potrebné vypočítať transformačnú maticu z množiny bodov, ktorých pozícia je známa v oboch obrazoch. Na presné definovanie transformačnej matice afinnej transformácie A sú potrebné 3 body, ktorých pozícia je známa v oboch obrazoch. Z daných bodov je potom možné vytvoriť sústavu šiestich rovníc o šiestich neznámych [25]:

$$\underbrace{p_x^1 = a_1 p_x^{1'} + a_2 p_y^{1'} + t_x, p_y^1 = a_4 p_x^{1'} + a_5 p_y^{1'} + t_y}_{p^1 \leftrightarrow p^{1'}}, \underbrace{p_x^2 = a_1 p_x^{2'} + a_2 p_y^{2'} + t_x, p_y^2 = a_4 p_x^{2'} + a_5 p_y^{2'} + t_y}_{p^2 \leftrightarrow p^{2'}}, \underbrace{p_x^3 = a_1 p_x^{3'} + a_2 p_y^{3'} + t_x, p_y^3 = a_4 p_x^{3'} + a_5 p_y^{3'} + t_y}_{p^3 \leftrightarrow p^{3'}}, \quad (3.22)$$

Ak existuje viac korešpondujúcich bodov ako 3 a ich pozície sú bezchybne určené, nezáleží na tom, ktorá trojica sa použije pri výpočte afinnej transformácie. Problém je, ak ich pozície sú určené s nejakou chybou. V tom prípade každá trojica dá iný výsledok. Existuje niekoľko metód pre minimalizáciu chyby riešenia, ako napríklad **metóda najmenších štvorcov**. Táto metóda spočíva v tom, že nájde takú transformáciu, ktorej suma štvorcov chýb bude čo najmenšia [22]. Takáto chyba má nasledujúci vzorec:

$$E_{LS} = \sum_i \|Ap'_i + \vec{t} - p_i\|^2 \quad (3.23)$$

Matica A a vektor \vec{t} popisujú afinnú transformáciu a $p_i \leftrightarrow p'_i$ sú všetky korešpondujúce body.

3.4 Párovanie korešpondujúcich bodov

Aby bolo možné nájsť transformáciu medzi dvoma obrazmi, je potrebné spárovať zhodné body nájdené v rôznych obrazoch. Jeden z možných algoritmov na nájdenie korešpondujúcich bodov medzi dvoma obrazmi je **algoritmus hrubej sily (brute force)**. Spočíva v postupnom prechádzaní bodov prvého obrazu. Vybraný bod z prvého obrazu je potom porovnaný s každým bodom z druhého obrazu. Na porovnávanie môžu byť použité metódy popísané v podkapitole 3.2. Ako korešpondujúci bod s daným bodom je určený ten bod, ktorý s ním má najvyššiu zhodu.

Táto metóda ale môže generovať falošné korešpondencie. Napríklad niektorý bod z prvého obrazu sa nemusí nachádzať v druhom obraze a tak bude k danému bodu priradený bod, ktorý mu je najviac podobný. Existujú však metódy, ktoré pracujú na úplne inom princípe ako porovnávanie okolia bodov a týmto problémom netrpia.

Jednou z takých metód je **RANSAC** (Random Sample Consensus – Konsenzus náhodných vzoriek) [7]. Je to metóda používaná pri odhadovaní matematických modelov z dátových množín a je schopná tolerovať násobne väčší počet chybných vzoriek ako správnych vzoriek. Iteratívne vyberá náhodné vzorky z dátovej množiny, podľa ktorých vytvára model a následne zisťuje, koľko vzoriek z celej dátovej množiny zodpovedá s nejakou maximálnou chybou tomuto modelu. Vzorky, ktoré vyhovujú danému modelu sa nazývajú *inliers*. Ak je *inliers* vzoriek dostatočný počet, je daný model zaradený medzi vhodné modely. Po danom počte iterácií je z vhodných modelov vybraný ten s najväčším počtom *inliers* vzoriek.

Pri hľadaní korešpondenčných bodov pre afinnú transformáciu tento algoritmus funguje tak, že v cykle vyberá vždy 3¹ náhodné páry bodov (pre urýchlenie sa používajú metódy na odstránenie nevhodných párov), vypočíta z nich afinnú transformáciu a zistí, koľko z ostatných párov bodov vyhovuje s nejakou maximálnou chybou tejto transformácii. Ak ich je dostatočný počet, daná transformácia je zaradená medzi vhodné transformácie. Po danom počte iterácií je z vhodných transformácií vybraná tá najlepšia a na základe nej všetky *inliers* páry bodov. Následne býva ešte daná transformácia prepočítaná (napríklad metódou najmenších štvorcov) s ohľadom na všetky *inliers* páry bodov.

Metóda RANSAC taktiež môže byť použitá už len na prefiltrovanie množín dopredu spárovaných bodov, ktoré môžu obsahovať nekorešpondujúce dvojice.

¹Pre analytické spočítanie afinnej transformácie sú potrebné práve tri páry bodov, kapitola 3.3

3.5 Prevzorkovanie obrazu

Pri aplikácii transformačnej matice na obraz nastáva problém. Digitálny obraz obsahuje vzorky len pre celočíselné súradnice, ale po vynásobení bodu maticou môžu vzniknúť ne celočíselné súradnice a obraz je potrebné prevzorkovať. **Prevzorkovanie** je proces prevodu diskretného signálu na iný diskretný signál [8]. Proces prevodu diskretného obrazu na spojitý sa nazýva **rekonštrukcia**. Existuje niekoľko rekonštrukčných algoritmov.

Najjednoduchším spôsobom rekonštrukcie je **interpolácia najbližším susedom**. Funguje na princípe použitia hodnoty najbližšieho pixelu v okolí:

$$I_N(x, y) = I_S(\lfloor x + 0.5 \rfloor, \lfloor y + 0.5 \rfloor) \quad (3.24)$$

I_N je spojitá funkcia nových hodnôt a I_S je pôvodná diskretná funkcia. Metóda je jednoduchá na výpočet avšak má niekoľko nežiadúcich efektov. Môže spôsobovať zvýraznenia alebo zmiznutia niektorých kontúr z obrazu, napríklad tenkých čiar.

Ďalšou metódou je **bilinéarna interpolácia**. Pri tejto metóde je nová hodnota vypočítaná zo štyroch najbližších susedných pixelov, ako ich vážený priemer vzhľadom na vzdialenosť od novej pozície.

$$\begin{aligned} I_N(x, \lfloor y \rfloor) &= (\lceil x \rceil - x)I_S(\lfloor x \rfloor, \lfloor y \rfloor) + (x - \lfloor x \rfloor)I_S(\lceil x \rceil, \lfloor y \rfloor) \\ I_N(x, \lceil y \rceil) &= (\lceil x \rceil - x)I_S(\lfloor x \rfloor, \lceil y \rceil) + (x - \lfloor x \rfloor)I_S(\lceil x \rceil, \lceil y \rceil) \\ I_N(x, y) &= (\lceil y \rceil - y)I_N(x, \lfloor y \rfloor) + (y - \lfloor y \rfloor)I_N(x, \lceil y \rceil) \end{aligned} \quad (3.25)$$

Prechody sú hladšie ako pri predchádzajúcej metóde, ale na druhej strane táto metóda rozostruje ostré hrany.

Často používanou metódou je **bikubická interpolácia**. Ako už názov napovedá, bikubická interpolácia využíva pri výpočte polynómy tretieho stupňa. Na výpočet hodnôt býva použitých viac hodnôt ako pri bilinearnej interpolácii, často to býva okolie 4×4 . Je viac spôsobov ako dosiahnuť bikubickú interpoláciu, napríklad Lagrangeove polynómy, kubické splajny alebo algoritmus kubickej konvolúcie.

Jadro Keysovej [13] kubickej konvolúcie pozostáva z polynómov tretieho stupňa definovaných na intervaloch $(-2, -1)$, $(-1, 0)$, $(0, 1)$ a $(1, 2)$. Mimo intervalu $(-2, 2)$ je jadro konvolúcie 0. Preto stačia pre vyhodnotenie štyri najbližšie hodnoty pôvodného obrazu. Jadro musí byť symetrické a má formu

$$u(s) = \begin{cases} 1 & s = 0 \\ \frac{3}{2}|s|^3 - \frac{5}{2}|s|^2 + 1 & 0 < |s| < 1 \\ 0 & |s| = 1 \\ -\frac{1}{2}|s|^3 + \frac{5}{2}|s|^2 - 4|s| + 2 & 1 < |s| < 2 \\ 0 & 2 < |s| \end{cases} \quad (3.26)$$

Pre interpoláciu obrazu musí byť jadro použité pre oba rozmery. Preto sa táto interpolácia nazýva bikubická. Vďaka použitiu polynómu vyššieho rádu a väčšieho okolia ako pri bilinearnej interpolácii, je bikubická interpolácia kvalitnejšia.

Existujú aj pokročilejšie metódy, ako napríklad Lanczosova interpolácia, ktorých výsledky sú ešte kvalitnejšie. Dané metódy využívajú ešte väčší počet susedných pixelov a zložitejšie aproximačné funkcie. Preto sú aj časovo náročnejšie.

Kapitola 4

Zhodnotenie problému a návrh parametrov riešenia

Táto kapitola sa zaoberá problémom nežiadúceho pohybu dopravných stereokamier. Prvá podkapitola rozoberá ako tento problém vzniká a čo môže spôsobiť a aj to, že v literatúre je len málo článkov zaoberajúcich sa touto témou. V druhej podkapitole sú predstavené parametre, ktorými som sa riadil pri implementácii.

4.1 Nepresnosť meraní spôsobená pohybom kamery

Pri používaní stereokamery na presné sledovanie pohybu objektov je dôležité, aby namerané výsledky boli čo najpresnejšie. To sa týka napríklad aj merania rýchlosti vozidiel dopravnou kamerou. Presnosť meraní ovplyvňuje okrem kalibrácie kamery a schopnosti určiť presnú polohu sledovaného objektu v obraze aj pozícia kamery voči pozícii, pre ktorú bola kamera skalibrovaná. Pri stereokamere môžu nastať dva typy posunov: stereokamera sa pohne ako celok a zmení sa pozícia súradnicovej sústavy kamery voči súradnicovej sústave sveta (kapitola 2.1), alebo sa kamery stereokamery pohnú voči sebe, čo spôsobí zmenu epipolárnej geometrie (kapitola 2.2). Oba prípady potom môžu spôsobovať chyby v meraní.

Týmto pohybom sa dá čiastočne zabrániť tak, že kamera bude pevne umiestnená na základni, ktorá sa zdanlivo nehýbe. Napríklad na konštrukcii mosta. Avšak aj takto pevné konštrukcie sa pohybujú. Či už vplyvom vetra, prejazdu ťažkého nákladného vozidla, ktoré konštrukciu rozvibrujú alebo vplyvom teplotnej rozťažnosti materiálu a starnutia materiálu. Práce zamerané na počítanie rýchlosti vozidiel pomocou stereokamery spomenuté v kapitole 2.4 tento problém v podstate neriešia. Kamera je umiestnená na nejakom relatívne stabilnom objekte (statív, stĺp...) a jej pohyby sa ignorujú. Pri výpočtoch rýchlosti sa predpokladá, že kamera sa vôbec nehýbe. To môže byť jedným z dôvodov, prečo napríklad v práci El Buziadyho *et al.* [5] je stredná kvadratická chyba pri počítaní rýchlosti 1.67 km/h pri rýchlostiach 60-90 km/h oproti rýchlosti nameranej pomocou GPS.

Z rozobratých prác zaoberajúcich sa presnosťou stereospracovania je práca Chung-Hua Chu [2] jediná, ktorá ponúka nejaké riešenie. Toto riešenie je ale zamerané úplne iný problém. Predstavená stabilizácia stereovidea je zameraná na zlepšenie kvality pozerania a nie zlepšenie presnosti meraní. Z tohoto dôvodu som sa rozhodol na podnet môjho vedúceho bakalárskej práce navrhnúť a implementovať algoritmus, ktorý bude tieto pohyby detekovať a kompenzovať.

4.2 Návrh parametrov, ktorými by sa mal kompenzačný algoritmus riadiť

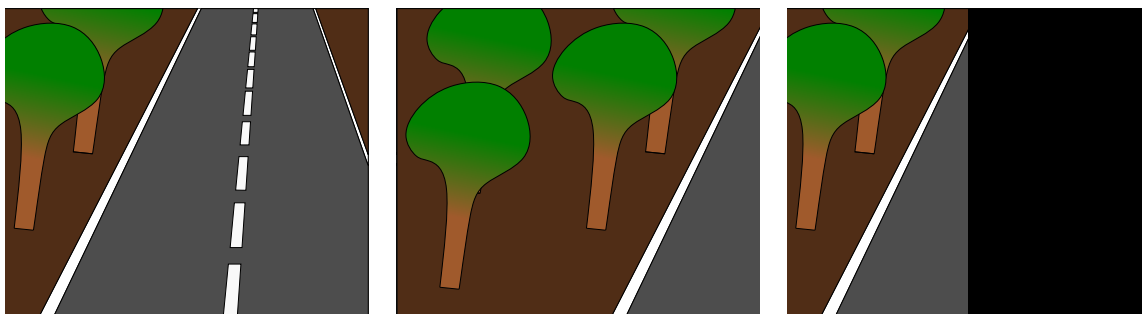
Cieľom algoritmu je zabezpečiť, aby pohyb obrazov obidvoch kamier stereokamery bol čo najmenší oproti východzej pozícii. Keďže obe kamery stereokamery sa môžu hýbať voči okoliu ale aj voči sebe, hľadať len spoločný pohyb dvojice kamier je nedostačujúce. Preto som sa rozhodol, že budem analyzovať pohyb každej kamery zvlášť.

Ak má algoritmus detekovať pohyb snímky, musí ho detekovať voči niečomu. Ideálne by bolo určiť pohyb snímky voči snímke, ktorá bola zaznamenaná, keď kamery boli v pozícii, pre ktorú boli skalibrované. Táto snímka bude ďalej označovaná ako referenčná. Referenčná snímka bude porovnávaná so všetkými ostatnými snímkami, preto by mala obsahovať len objekty, ktoré sa budú nachádzať aj na všetkých ostatných snímkach. Nemala by obsahovať napríklad prechádzajúce vozidlá. Keďže postup automatického zisťovania, či sa na snímke nachádzajú vozidlá, pohybujúce alebo stojace, alebo iné objekty, ktoré budú meniť v priebehu času svoju pozíciu by bol netriviálny, rozhodol som sa, že určovanie referenčnej snímky bude prebiehať manuálne.

Na snímkach sa ale aj tak môžu objaviť objekty, ktoré síce z obrazu nezmiznú, ale môžu meniť svoj tvar. Môže sa jednať o stromy alebo iné rastliny. Tie totiž budú časom rásť, budú sa hýbať s náporom vetra, môže im opadať lístie alebo môžu nastať iné zmeny. Keby sa podľa týchto objektov určoval pohyb snímok, mohol by byť vďaka ich zmenám chybný. Preto je potrebné v referenčnej snímke manuálne označiť oblasti, ktoré nebudú použité na hľadanie pohybu.

Aby bol algoritmus použiteľný, musí byť dostatočne rýchly. Ak by zvládol spracovať maximálne jednu snímku za sekundu, aj keby bol úplne dokonalý, bol by zbytočný, lebo by výrazne spomaľoval ďalšie spracovanie snímok.

Ďalší z parametrov algoritmu je, aké veľké pohyby má byť schopný korigovať. Keďže tento algoritmus má byť použitý na minimalizovanie malých pohybov, je zbytočné, dokonca nežiadúce, aby algoritmus stabilizoval snímky, ktoré sú napríklad o polovicu posunuté oproti pôvodnej pozícii. V takomto prípade by stabilizácia bola zbytočná, keďže kamera by už nesnímala tú časť scény, ktorú pôvodne mala. Tento problém je znázornený na obrázku 4.1. Ďalej sa dá predpokladať, že pohyb pevne umiestnenej kamery len vplyvom prírodných živlov a prechádzajúcich vozidiel nebude príliš veľký. Preto som sa rozhodol, že stačí, aby algoritmus korigoval pohyby maximálne v jednotkách pixelov.



Obr. 4.1: Referenčná snímka (vľavo), snímka posunutá o pol snímky voči referenčnej snímke (v strede), kompenzovaná posunutá snímka (vpravo)

Keďže pohyby kamery nie sú len v celých pixeloch, tak takisto vyhodnocovanie pohybov by nemalo byť len v celých pixeloch. Algoritmus preto musí byť schopný pracovať na subpixelovej úrovni.

Ďalší problém je, ako popísať transformáciu medzi referenčnou snímku a ostatnými snímkami. Ideálna transformácia na popis pohybu kamery by bola perspektívna transformácia, ale keďže sa bude jednať len o malé posuny kamery, perspektíva sa zmení iba minimálne a tento pohyb stačí vyjadriť pomocou afinnej transformácie, ktorá je jednoduchšia na počítanie.

Posledným problémom je, pomocou akých prostriedkov algoritmus implementovať. Ponúkajú sa dve možnosti: využiť programovací jazyk a systém priamo zameraný na matice a výpočty, napr. MATLAB¹ alebo GNU Octave² alebo využiť nejaký bežný programovací jazyk spolu s knižnicou zameranou na spracovanie obrazu. Takou knižnicou je napríklad OpenCV (Open Source Computer Vision Library)³.

Rozhodol som sa pre knižnicu OpenCV a programovací jazyk C++. Jednou z výhod bolo, že s oboma už som v minulosti pracoval, preto som nemusel začínať úplne odznovu a hľadať v manuáloch, ako vôbec nainštalovať OpenCV a naprogramovať jednoduchý program. Ďalšie dôvody boli, že OpenCV má priamo implementované niektoré algoritmy z kapitoly 3, je multiplatformová a je open source. Jazyk C++ bol zvolený tiež preto, aby potom výsledný algoritmus mohol byť jednoducho použiteľný v iných aplikáciách, čo uľahčuje modularita C++ a je jednoduché v ňom využiť knižnicu OpenCV a nie je problém s rozhraním knižnice, keďže aj OpenCV je napísaná v C++.

Zhrnutie vlastností, ktoré by mal mať kompenzačný algoritmus a parametrov, ktorými by sa mal riadiť:

- zlepšenie presnosti metód na meranie rýchlosti pomocou stereokamery
- použitý programovací jazyk C++,
- knižnica využitá na spracovanie obrazu OpenCV,
- stereosnímky spracovávané jednotlivo,
- stabilizovanie voči konkrétnej snímke (referenčná snímka),
- možnosť manuálne odstrániť nechcené oblasti zo snímky,
- priemerná rýchlosť spracovania aspoň 30 stereosnímkov za sekundu,
- odhalenie pohybov aspoň do vzdialenosti 10 pixelov,
- popis pohybu afinnou transformáciou,
- detekcia pohybu na subpixelovej úrovni.

Týmito bodmi som sa riadil pri následnej implementácii. Implementácia programu je popísaná v nasledujúcej kapitole.

¹<https://www.mathworks.com/products/matlab.html>

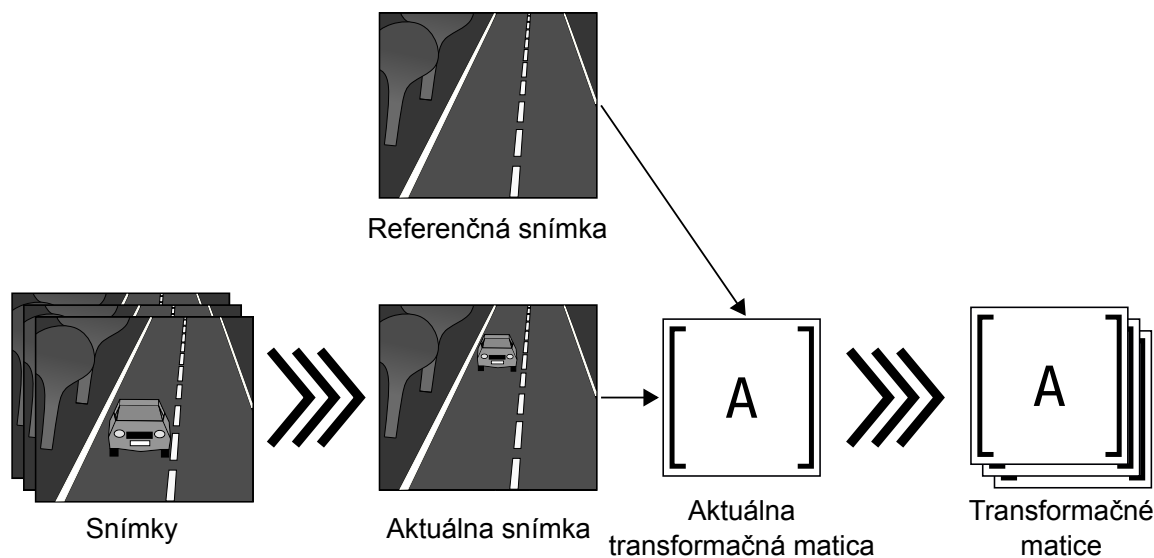
²<https://www.gnu.org/software/octave/index>

³<https://opencv.org/>

Kapitola 5

Implementácia navrhnutého riešenia

Táto kapitola sa zaoberá implementáciou navrhnutého riešenia na stabilizovanie snímok. Na začiatku kapitoly je predstavené zjednodušené fungovanie algoritmu ako celku. V podkapitolách sú potom podrobne rozobrané jednotlivé časti algoritmu. Prvá podkapitola sa venuje spracovaniu referenčnej snímky. V druhej podkapitole je rozobraný základný algoritmus na nájdenie korešpondujúcich bodov medzi aktuálne spracovávanou snímkou a referenčnou snímkou. V tretej podkapitole je popísaný algoritmus, ktorý je použitý v prípade, keď základný algoritmus nedokáže nájsť dostatočný počet korešpondujúcich bodov. V poslednej podkapitole je predstavený spôsob porovnávania korešpondujúcich bodov a následný výpočet transformačnej matice afinnej transformácie.



Obr. 5.1: Zjednodušená schéma fungovania algoritmu na zisťovanie posunu snímok

Na obrázku 5.1 je možné vidieť princíp fungovania celého programu. Na začiatku behu programu je načítaná referenčná snímka. Potom je sekvenčne načítavaná postupnosť snímok určených na stabilizovanie. Po načítaní je každá snímka porovnaná s referenčnou snímkou. Porovnávajú sa pozície rohov z referenčnej snímky a ostatných snímok pomocou korelácie výrezov s rohmi (proces porovnávania je podrobne vysvetlený v nasledujúcich podkapitolo-

lách). Z výsledkov porovnania jednotlivých snímok je vypočítaná matica afinnej transformácie. Výstupom programu sú matice afinnej transformácie, pre každú načítanú snímku jedna.

5.1 Spracovanie referenčnej snímky

Pred samotným spustením programu a spracovaním referenčnej snímky je potrebné vykonať dve manuálne činnosti. Vyhľadať vhodnú referenčnú snímku a vytvoriť masku na odmaskovanie jej nevhodných častí. Referenčná snímka by mala obsahovať čo najviac pozadia, ktoré sa v čase nemení a čo najmenej pohyblivých objektov. Taktiež by mala byť zaznamenaná v čase, keď kamera bola vo svojej východzej pozícii, pre ktorú bola skalibrovaná.

Ak aj napriek tomu sú na referenčnej snímke objekty, ktoré sa môžu hýbať (napríklad stromy vo vetre), je potrebné ich odmaskovať. Na vytvorenie masky je možné použiť buď priložený program alebo ľubovoľný grafický editor. Priložený program umožní užívateľovi otvoriť vybranú referenčnú snímku a pomocou myši označiť nechcené oblasti snímky. Program následne exportuje masku referenčnej snímky vo formáte PNG. Príklad masky možno vidieť na obrázku 5.2. Masky vytvorené inými programami musia mať rovnaké rozlíšenie ako referenčná snímka, musia byť v odtieňoch šedej a pixely v nechcených oblastiach musia byť čierne (musia mať hodnotu 0). Všetky ostatné hodnoty pixelov označujú chcené oblasti. Ak pri spustení programu nie je zadaná žiadna maska, automaticky sa vytvorí prázdna maska (bez čiernych oblastí).

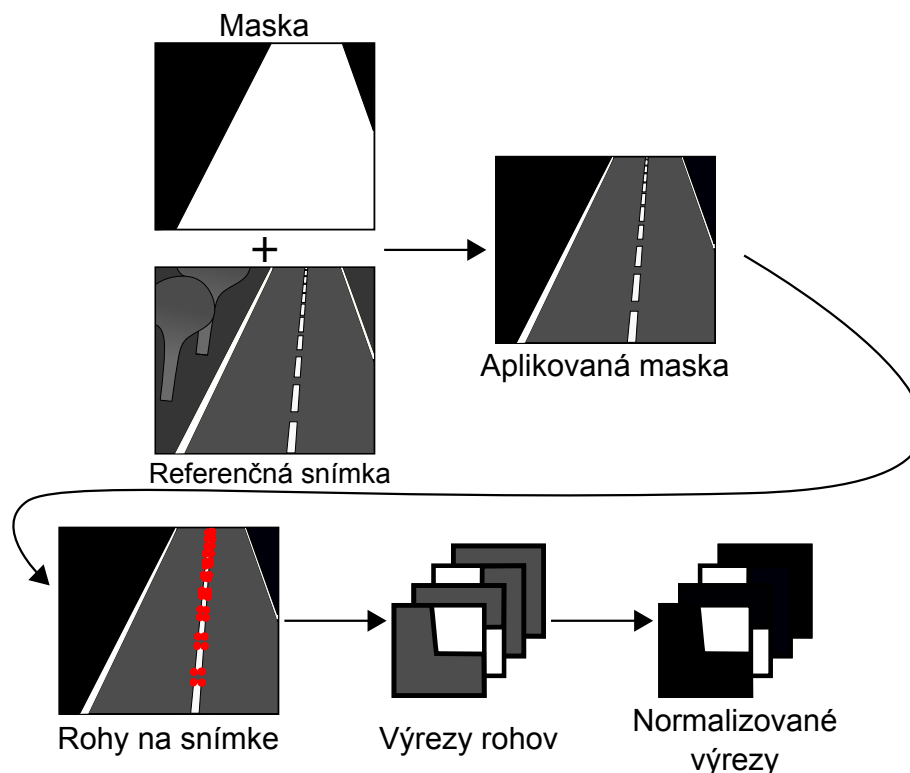


Obr. 5.2: Snímka programu na vytvorenie masky (vľavo) a vytvorená maska vo formáte PNG (vpravo). V programe sú nechcené oblasti označené červenou farbou a v maske čiernou.

Program referenčnú snímku a jej masku po načítaní začne spracovávať. Tento proces je znázornený na obrázku 5.3. Spracovanie referenčnej snímky pred začatím porovnávania je dôležitý krok. Celý proces je vďaka nemu podstatne rýchlejší. Všetky výpočty na referenčnej snímke sú urobené iba raz a nie pri každom porovnávaní.

Referenčná snímka je načítaná ako šedotónový obrázok. Na načítaný obrázok je následne aplikovaná maska. Vďaka maske nie sú na hľadanie posunu snímok voči referenčnej snímke použité nevhodné oblasti.

Pri hľadaní posunu sa nevyužíva celá nezamaskovaná oblasť referenčnej snímky. Používa sa len istý počet rohových bodov. Na vyhľadanie rohov v referenčnej snímke je použitý Shi-



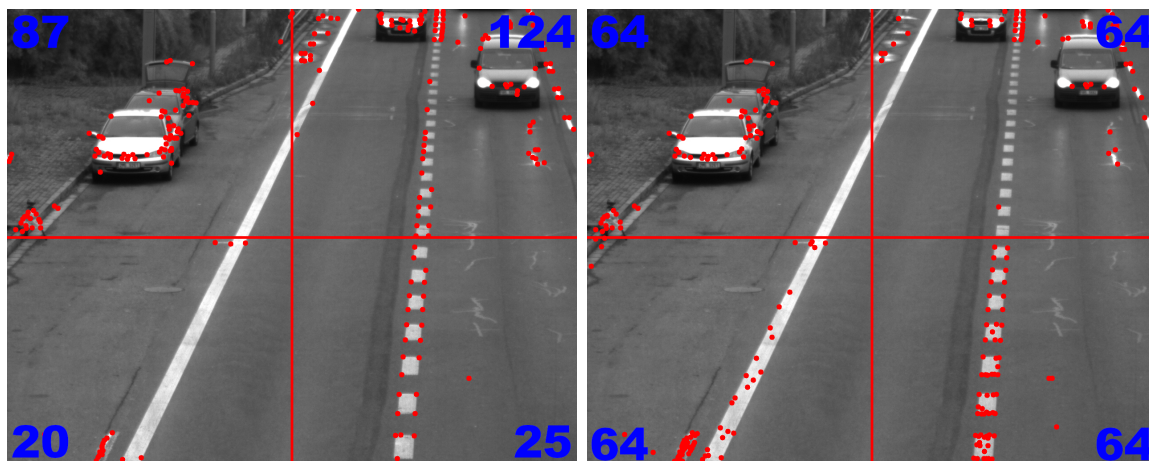
Obr. 5.3: Schéma spracovania referenčnej snímky: Po načítaní referenčnej snímky sa na snímku aplikuje maska. Následne sa na snímke v nemaskovaných oblastiach vyhľadajú rohové body. Potom sa spravia výrezy zo snímky so stredmi v nájdených rohoch. Tieto výrezy sú následne normalizované.

Tomasi detektor (spomenutý v kapitole 3.1). Detektor rohov je použitý namiesto detektora vlastností (detektory založené na podobnom princípe ako SIFT, kapitola 3.1) preto, lebo sa predpokladá, že hlavný pohyb snímok bude posun a nie zmena veľkosti. Detekcia rohov je výpočtovo jednoduchšia a posun jej nerobí problém.

Rohy sú vybrané podľa kvality (hodnota R Shi-Tomasi detektora). Problém tohto riešenia je v tom, že sa môže stať, že väčšina rohov bude v jednej časti obrazu, tak ako je to možné vidieť na obrázku 5.4. To môže spôsobiť problém, ak by tú časť v niektorom momente prekrylo nejaké vozidlo. Podstatne by sa znížil počet korešpondujúcich rohov medzi referenčnou snímkou a snímkou s vozidlom čo prekryva oblasť s najväčším počtom rohov. Aby som tomuto zamedzil, rohy sú vyhľadané nezávisle pre každý kvadrant snímky. To zaručí rovnomernejšie rozloženie rohov po ploche snímky a zníži riziko prekrytia väčšiny rohov vozidlami.

Nájdené pozície rohov sú však len na pixelovej úrovni. Pre lepšiu presnosť je následne použitý algoritmus na presnejšie určenie pozície rohov na subpixelovej úrovni založený na princípe ortogonálnych vektorov popísaný v kapitole 3.1. Tento algoritmus vezme pozíciu rohu získaného pomocou Shi-Tomasi detektora a vráti presnejšiu subpixelovú pozíciu toho istého rohu.

V ďalšom kroku sú spravené výrezy rohov a ich blízkeho okolia. Keďže roh má subpixelové súradnice, aby roh mohol byť v strede svojho výrezu, je potrebné, aby bol aj výrez



Obr. 5.4: Vyhľadanie 256 najlepších rohov v celom obrázku (vľavo) a vyhľadanie 64 najlepších rohov v každom kvadrante obrázka (vpravo). Čísla v rohoch označujú počet nájdených rohov v danom kvadrante. Červené bodyk označujú nájdené rohy.

spravený na subpixelovej úrovni. Preto je potrebné výrez prevzorkovať. Na to je použité bilinéárne prevzorkovanie (kapitola 3.5).

V každom takomto výreze sú potom všetky pixely normalizované. Normalizácia je dôležitá, pretože počas dňa sa mení osvetlenie snímanej cesty a teda aj osvetlenie rohov, nenormalizované výrezy inak nasvieteného rohu budú odlišné, ale ich normalizované verzie budú podobné.

Keď je spracovaná referenčná snímka, začnú sa postupne spracovávať ďalšie snímky. Na spracovanie ďalších snímok má program dva algoritmy. Tie sú vysvetlené v nasledujúcich dvoch podkapitolách.

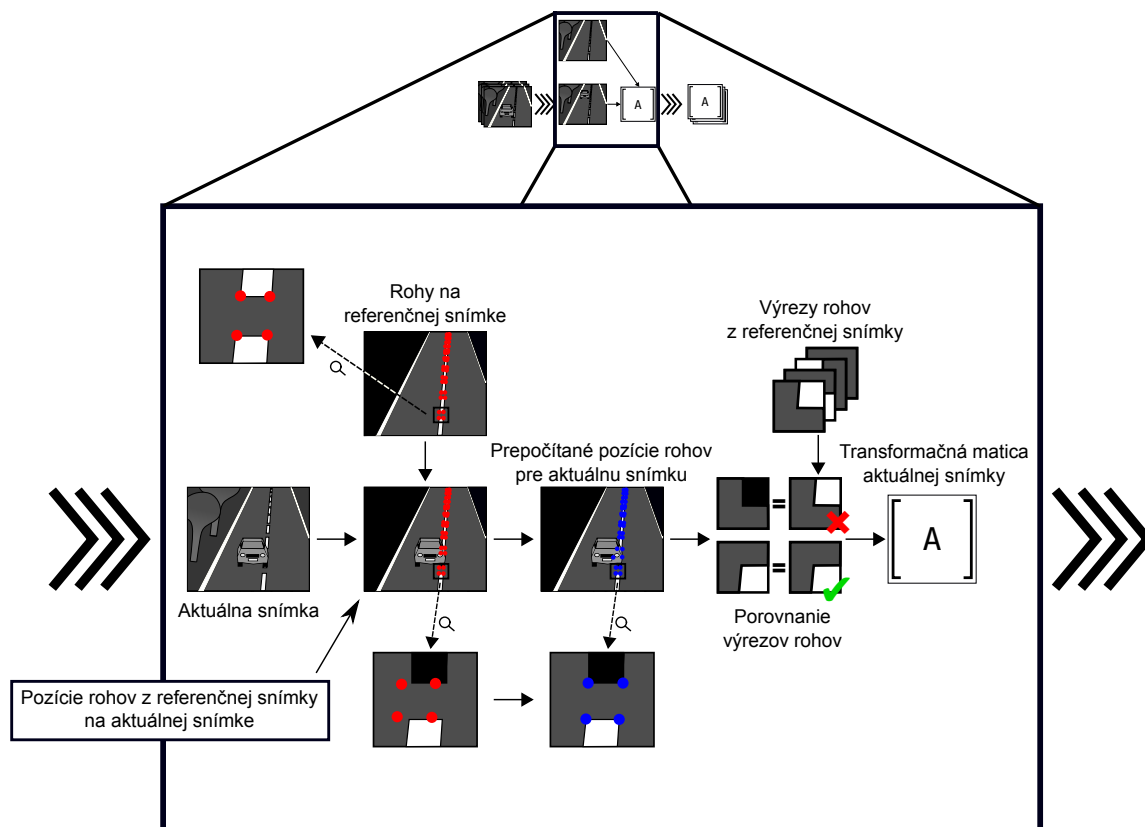
5.2 Rýchly algoritmus na hľadanie korešpondujúcich rohov

Tento algoritmus je predvolený pri hľadaní korešpondujúcich bodov medzi referenčnou snímkou a práve načítanou snímkou. Jeho schému je možné vidieť na obrázku 5.5. Algoritmus vychádza z predpokladu, že pohyb snímok je vo väčšine prípadov minimálny a rohové body nájdené na referenčnej snímke majú podobné súradnice ako tie isté rohy na aktuálne spracovávanej snímke. Preto nie je potrebné prechádzať celú aktuálnu snímku detektorom rohov, ale stačí mierne upraviť pozíciu rohov z referenčnej snímky. Na to je použitý algoritmus na zistenie subpixelovej pozície rohu, ten istý algoritmus ako algoritmus použitý pri spracovaní referenčnej snímky. Ten nájde najsilnejší roh v okolí so subpixelovou presnosťou. Veľkosť prehľadávaného okolia je 3×3 pixely. Nájdený roh môže byť:

- rovnaký roh ako roh na referenčnej snímke,
- roh na objekte, ktorý prekryl pôvodný roh,
- úplne iný roh, z dôvodu príliš veľkého posunutia snímky voči referenčnej snímke.

Preto sa následne spravia výrezy rohov so subpixelovou presnosťou a tieto výrezy sa potom normalizujú. V ďalšom kroku je každý výrez porovnaný so svojim korešpondujúcim výrezom z referenčnej snímky. Ako porovnávací algoritmus je použitá korelácia. Ak je výsledok

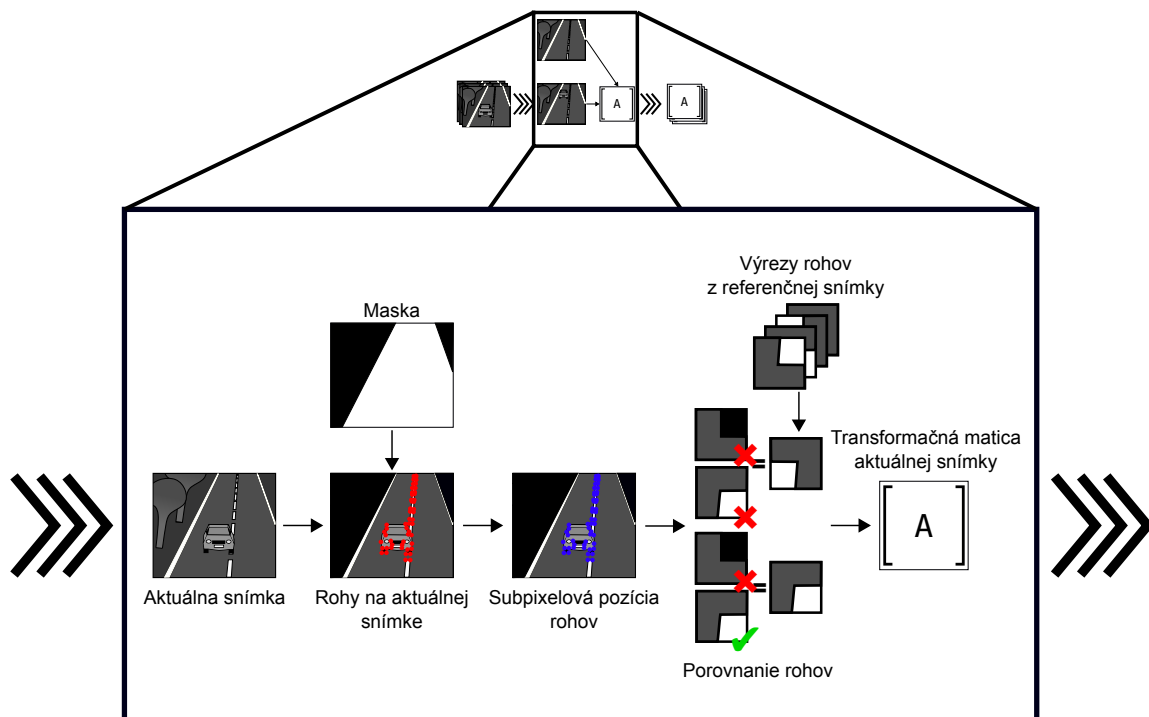
korelácie vyšší ako určená hranica, čo znamená, že nájdený roh je pravdepodobne zhodný s korešpondujúcim rohom z referenčnej snímky, tak sa korešpondujúce rohy použijú pri výpočte transformačnej matice. Ak je výsledok nižší, jedná sa pravdepodobne o iný roh a preto sa tento roh zahodí. Ak nastane situácia, že sa nenájde dostatočný počet korešpondujúcich rohov (napríklad z dôvodu veľkého pohybu aktuálnej snímky), nie je možné vypočítať transformáciu. V takomto prípade je použitý robustnejší algoritmus na hľadanie korešpondujúcich rohov, ktorý je popísaný v nasledujúcej podkapitole.



Obr. 5.5: Schéma rýchleho algoritmu na hľadanie korešpondujúcich rohov: Na aktuálnej snímke sú v okolíach pozícií rohov z referenčnej snímky dohľadané presné pozície rohov. Výrezy rohov sú porovnané s výrezmi z referenčnej snímky. Rohy vyhodnotené ako zhodné sú následne použité na výpočet transformačnej matice.

5.3 Robustnejší algoritmus hľadania korešpondujúcich rohov

Robustnejší algoritmus je použitý až keď rýchly algoritmus zlyhá. Je určený na nájdenie väčších posunov. Pri hľadaní rohov už nevyužíva súradnice rohov z referenčnej snímky. Namiesto toho aplikuje na aktuálnu snímku masku z referenčnej snímky a vyhľadá v nej rohy pomocou Shi-Tomasi detektora. Je nutné vyhľadať viac rohov ako na referenčnej snímke, keďže môže nastať situácia, že na snímke bude zachytené vozidlo s veľa kvalitnými rohmi a tak by neboli nájdené rovnaké rohy ako na referenčnej snímke. Pozície rohov sú tak ako aj v referenčnej snímke prepočítané na subpixelovej úrovni.



Obr. 5.6: Robustnejší algoritmus na hľadanie korešpondujúcich rohov: Na aktuálnej snímke sú po aplikovaní masky nájdené rohy. Každý roh z referenčnej snímky je následne porovnaný s rohmi z aktuálnej snímky, ktoré sa nachádzajú v blízkosti súradníc daného rohu. Rohy vyhodnotené ako zhodné sú následne použité na výpočet transformačnej matice.

Ďalej sú spravené výrezy rohov taktiež na subpixelovej úrovni, ktoré sú potom normalizované. Nasleduje porovnávanie rohov. V tomto prípade už nestačí porovnať každý výrez z aktuálnej snímky s výrezom z referenčnej snímky, lebo nie je známe, ktorý roh z aktuálnej snímky by mal predstavovať roh na aktuálnej snímke. Preto je každý roh z referenčnej snímky porovnaný s rohmi z aktuálnej snímky, ktorých súradnice nie sú vzdialené od súradníc rohu na referenčnej snímke o viac ako je stanovená hodnota.

Medzi dvojice rohov, z ktorých sa počíta transformácia je zaradená len tá dvojica, ktorej výsledok vzájomnej korelácie normalizovaných výrezov je vyšší ako určená hranica a zároveň je vyšší ako výsledok korelácie dvojíc, ktorých súčasťou je rovnaký roh z referenčnej snímky.

5.4 Výpočet transformačnej matice afinnej transformácie

Na výpočet transformačnej matice sú použité spárované rohové body z aktuálnej a referenčnej snímky získané pomocou jedného z algoritmov popísaných v predchádzajúcich kapitolách. Ešte pred samotným počítaním sú tieto dvojice prefiltrované. Ak je nájdený veľký počet dvojíc, sú vybrané len tie s najlepšimi výsledkami korelácie. Ak ich je menej ako daná hranica (druhá korelačná hranica), použijú sa všetky. V prípade, keď ich je príliš málo, tak transformácia nie je vypočítaná. Pri príliš malom počte dvojíc totiž hrozí, že vypočítaná transformačná matica bude nesprávna. Dvojice bodov sú ešte na záver prefiltrované algoritmom RANSAC (kapitola 3.4) a pre výpočet matice je následne použitá metóda najmenších štvorcov (kapitola 3.3).

Kapitola 6

Testovanie implementovaného riešenia

Táto kapitola sa zaoberá testovaním implementovaného riešenia. Na testovanie boli použité snímky zo stereokamery vytvorené pre účely práce Najmana a Zemčíka [16]. Kamery danej stereokamery boli umiestnené na nosníku zo zliatiny hliníka vo vzdialenosti jeden meter od seba. Moje testovanie prebiehalo v troch fázach. V prvej fáze bola na testovanie použitá snímka z danej práce, ktorá bola umelo posúvaná a program mal odhaliť jej posunutie. Výsledky z tohto merania boli použité na hľadanie ideálnych parametrov pre algoritmu. V druhej fáze bol program otestovaný na celej sade snímok a v tretej fáze bolo testované či môj program pomôže zlepšiť presnosť programu na počítanie rýchlosti vozidiel zo stereosnímok.

6.1 Hľadanie optimálnych parametrov

V prvej fáze testovania som vybral náhodne jednu snímku zo súboru testovacích snímok. Tá bola v tejto časti testovania použitá ako referenčná snímka. Z nej som si vytvoril dve série snímok približne po tisíc kusov, ktoré boli rôzne posunuté oproti pôvodnej snímke. Prvá séria bola zameraná na testovanie rýchleho algoritmu, preto sa v nej nachádzali snímky s malým posunom (0 až 3 pixely). V druhej sérii sa nachádzali snímky na testovanie robustnejšieho algoritmu. Ich posuny boli preto väčšie (3 až 10 pixelov). Na posunutie bola použitá bikubická interpolácia. Obe série snímok som nechal opakovane spracovávať svojim programom, v ktorom som postupne menil jednotlivé vnútorné parametre, napríklad počet vyhľadaných rohových bodov a druhá korelačná hranica.

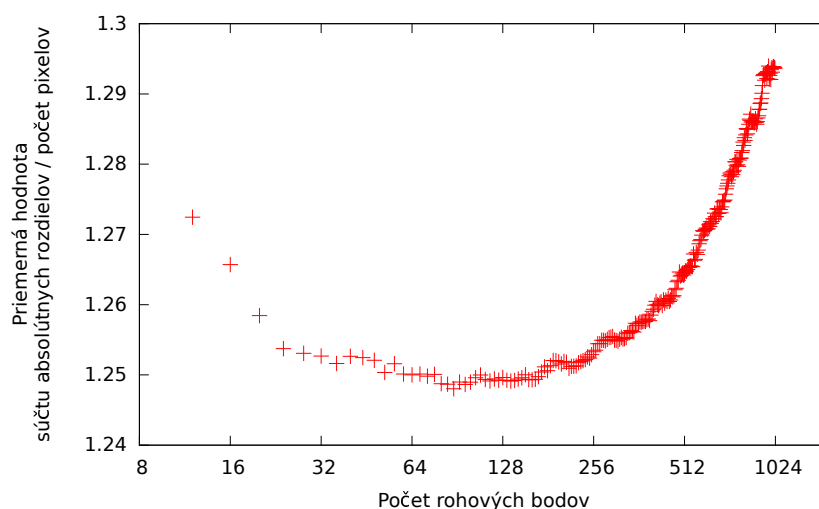
Výsledkom každého behu programu bola séria transformačných matíc. Tie som následne použil na transformáciu vytvorenej série posunutých snímok. V ideálnom prípade by výsledkom transformácie bola snímka identická s referenčnou snímkou. V skutočnosti snímka po transformácii vykazovala rozdiely. Na vyhodnotenie veľkosti odchýlok bola použitá metóda absolútnych rozdielov podľa vzorca:

$$E = \frac{\sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} |R_{i,j} - A_{i,j}|}{XY}, \quad (6.1)$$

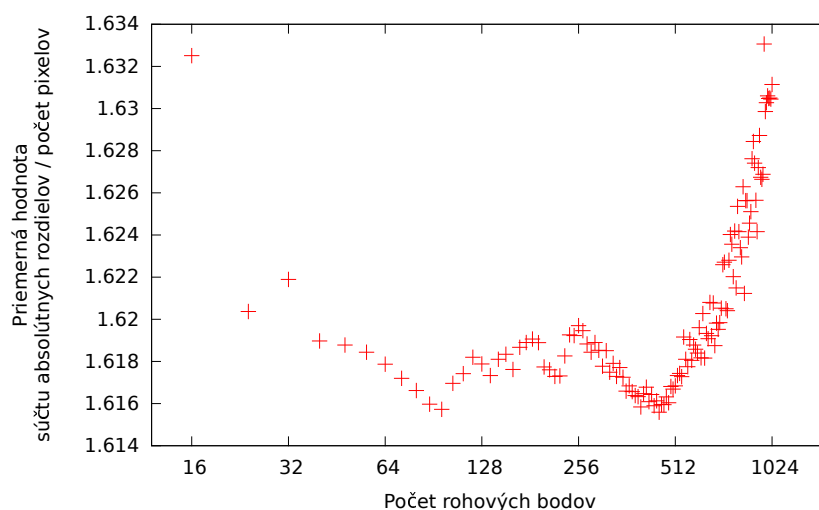
kde X a Y sú horizontálne a vertikálne rozlíšenie snímky. $R_{i,j}$ je intenzita pixelu referenčnej snímky na súradniciach (i, j) . $A_{i,j}$ je intenzita pixelu posunutej a transformovanej snímky podľa matice vypočítanej programom. Hodnota E predstavuje strednú absolútnu chybu

medzi referenčnou snímkou a transformovanou snímkou. Čím nižšia je hodnota E tým presnejšia je transformačná matica. Následne bola vypočítaná priemerná chyba zo všetkých snímok jedného behu programu.

Podľa tejto chyby som posudzoval hodnoty jednotlivých parametrov. Prvý parameter, ktorý som posudzoval, bol počet vyhľadávaných rohov na referenčnej snímke. Od tohto počtu sa odvíja aj počet vyhľadávaných bodov na ostatných snímkach pri použití robustnejšieho algoritmu. Ten je vždy dvojnásobný, aby bolo zaručené, že sa nájde na snímkach čo najväčší počet referenčných rohov, keďže na ostatných snímkach sa môžu nachádzať aj iné a kvalitnejšie rohy ako na referenčnej snímke. Výsledky tohto testovania boli zaznamenané pre prvú sériu do grafu 6.1 a pre druhú sériu do grafu 6.2.



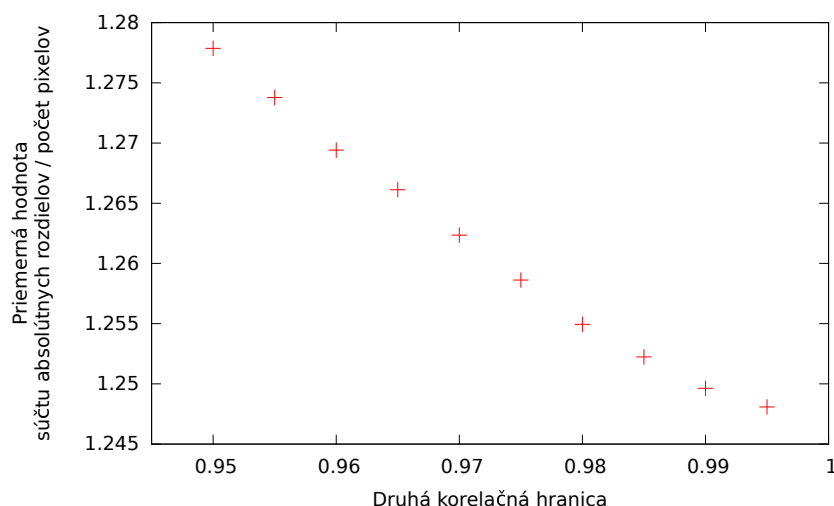
Obr. 6.1: Graf zobrazujúci priemernú veľkosť chyby vzhľadom na počet vyhľadávaných bodov pre prvú sériu snímok



Obr. 6.2: Graf zobrazujúci priemernú veľkosť chyby vzhľadom na počet vyhľadávaných bodov pre druhú sériu snímok

Najnižšia chyba pre prvú sériu má hodnotu 1.248 a bola dosiahnutá s 88 rohovými bodmi. Najnižšia chyba pre druhú sériu má hodnotu 1.6156 a bola dosiahnutá so 456 rohovými bodmi. Nízke hodnoty chyby pre druhú sériu boli dosiahnuté taktiež s počtami rohov 88 a 96 a to 1.6159 a 1.6157. Z oboch sérií som potom zvolil ako východziu hodnotu počtu rohov hodnotu 88 pre ďalšie testovanie. Po absolvovaní testu popísaného v nasledujúcej podkapitole bola táto hodnota zvýšená na 128, pretože s pôvodným parametrom vychádzalo veľa snímok, pre ktoré sa nespočítala transformačná matica pre nedostatok korešpondujúcich bodov.

Ďalší z testovaných parametrov bola druhá korelačná hranica. Výsledky tohto testovania je možné vidieť v grafoch 6.3 a 6.4. Z grafov je možné vidieť, že čím vyššia bola druhá korelačná hranica, tým nižšia bola chyba. Pôvodná hodnota tohto parametra bola preto nastavená na 0.995, ale po začatí testovania na reálnych snímkach bola znížená na hodnotu 0.99, keďže hodnoty korelácie na reálnych snímkach len vo výnimočných prípadoch prekročili hranicu 0.995.

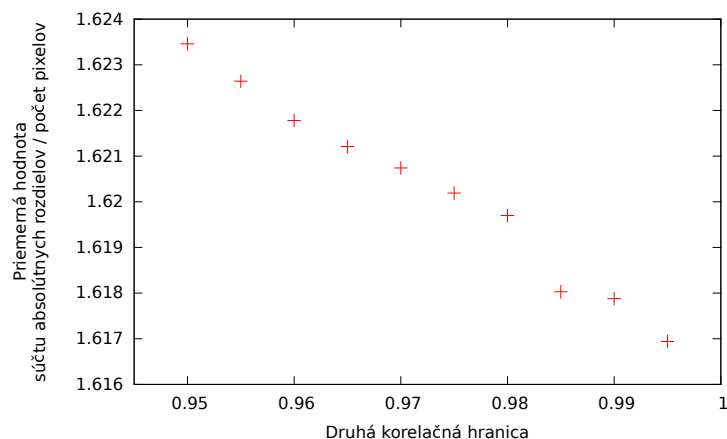


Obr. 6.3: Graf zobrazujúci priemernú veľkosť chyby vzhľadom na druhú korelačnú hranicu pre prvú sériu snímok

6.2 Problémy odhalené počas testovania

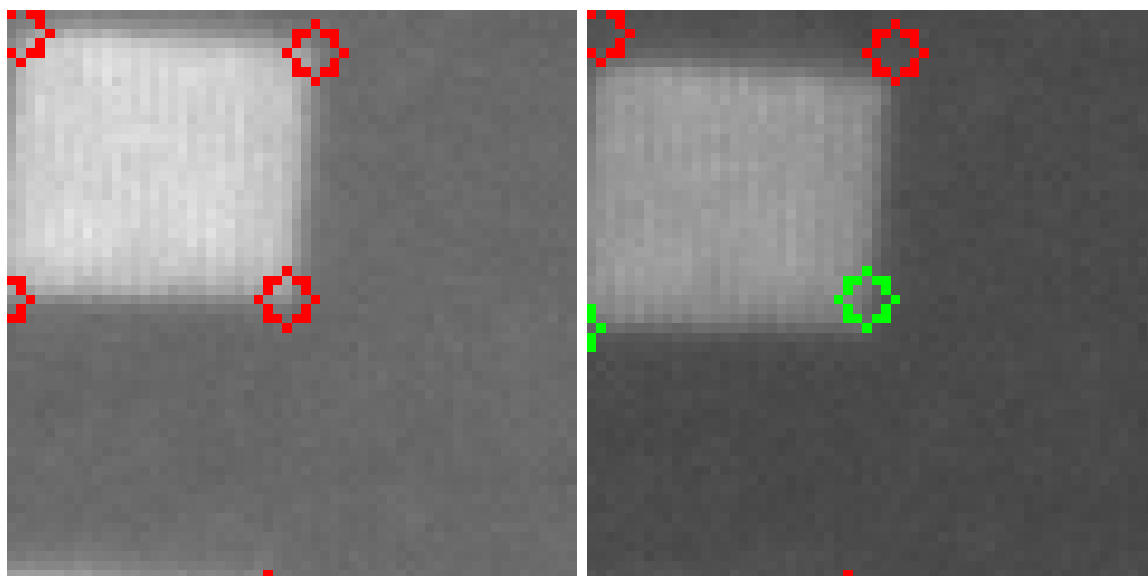
V ďalšej fáze testovania som už použil celú sadu testovacích snímok. Jedná sa o snímky vozidiel prechádzajúcich po cestnej komunikácii zaznamenané stereokamerou počas doby asi 45 minút. Zo snímok každej kamery som vybral vhodnú referenčnú snímku, ku ktorej som si vytvoril masku. S týmito referenčnými snímkami som spustil program pre každú kameru zvlášť. Cieľom tohto testovania bolo odhaliť nedostatky použitého algoritmu. Zameral som sa na matice „vytŕčajúce z davu“ (translačné parametre afinnej transformácie sa výrazne odlišovali od susedných matic). Snímky prislúchajúce týmto neštandardným maticiam som si vybral na samostatné spracovanie. Pri každej z týchto snímok som si nechal zobrazíť body, ktoré boli použité pri výpočte jej transformačnej matice. Vizuálnou kontrolou týchto bodov na snímkach som odhalil nedokonalosti algoritmu.

Jeden z odhalených problémov bol, že v niektorých prípadoch bol roh označený ako korešpondujúci aj keď nemal byť. Tento problém je možné vidieť na obrázku 6.5. Zelený bod sa



Obr. 6.4: Graf zobrazujúci priemernú veľkosť chyby vzhľadom na druhú korelačnú hranicu pre druhú sériu snímok

nenachádza na rovnakom rohu ako červený bod z referenčnej snímky a aj napriek tomu bol použitý pri výpočte transformačnej matice. Táto situácia nastala preto, lebo reálna pozícia rohu bola vzdialená o viac ako 3 pixely (rýchly algoritmus prehľadáva okolie troch pixelov) od referenčnej pozície a tak ju rýchly algoritmus nemohol nájsť, ale zároveň vzájomná korelácia výrezov týchto dvoch bodov bola 0.922. To je viac ako 0.9 čo bola pôvodná hranica na určovanie korešpondencie dvoch rohov. Aby sa zabránilo použitiu takýchto chybných dvojíc, bola hranica korešpondujúcich rohov zvýšená na 0.95.



Obr. 6.5: Výrez z referenčnej snímky (vľavo) s označenými referenčnými rohmi (červená) a výrez rovnakého miesta zo snímky s podozrivou transformačnou maticou (vpravo), kde sú pozície rohov upravené rýchlym algoritmom. Zelené body boli označené ako korešpondujúce body s referenčnými bodmi.

Ďalší z problémov, na ktoré som narazil, boli matice vypočítané z malého počtu korešpondujúcich bodov. Priemerná hodnota počtu bodov, z ktorých boli spočítané matice

je 65.2. Boli tam však matice, ktoré boli spočítané len z deviatich bodov. Niektoré z nich boli na prvý pohľad nesprávne. Väčšinou sa tých pár korešpondujúcich bodov nachádzalo v jednej časti snímky. Ak sa nejaký korešpondujúci bod nachádzal v inej časti, tak ho väčšinou algoritmus RANSAC označil za nevhodný. Na zamedzenie tohto problému boli pridané hranice na minimálny počet korešpondujúcich bodov. Ak tento počet nie je dosiahnutý, matica nie je vypočítaná, lebo by pravdepodobne aj tak bola chybná. V rámci tejto fázy testovania som do algoritmu pridal nasledujúce podmienky:

- ak rýchly algoritmus nenájde aspoň štvrtinu korešpondujúcich bodov z celkového počtu rohových bodov, tak je neúspešný a použije sa robustnejší algoritmus,
- ak robustnejší algoritmus nenájde aspoň osminu korešpondujúcich bodov z celkového počtu rohových bodov, transformačná matica nie je vypočítaná
- na výpočet matice je použitá aspoň štvrtinu korešpondujúcich bodov z celkového počtu rohových bodov (v prípade, že ich bolo nájdených menej, tak všetky nájdené korešpondujúce body), aj v prípade, že nemajú dostatočne vysokú hodnotu korelácie na dosiahnutie druhej korelačnej hranice.

Použité hodnoty boli empiricky odvodené tak, aby bolo čo najmenej vyslovene zle spočítaných transformácií. Upravenie týchto hodnôt spôsobí zmenu len na malej časti transformačných matíc, keďže väčšina matíc bola spočítaná z dostatočného počtu korešpondujúcich bodov. Po aplikovaní týchto pravidiel sa podstatne znížil počet výrazne odlišných matíc. Správnosť algoritmu som si overoval aj vizuálne. Vytvoril som video z postupnosti dvojíc snímok (vstupná snímka a snímka transformovaná vypočítanou maticou). Aby na videu bola viditeľná funkčnosť algoritmu nepoužil som celé snímky, ale len zväčšené výrezy stredovej čiary na ceste, na ktorej je viditeľný pohyb kamery. Vizuálnym hodnotením videa som dospel k záveru, že transformované snímky vykazovali výrazne lepšiu stabilitu ako vstupné snímky.

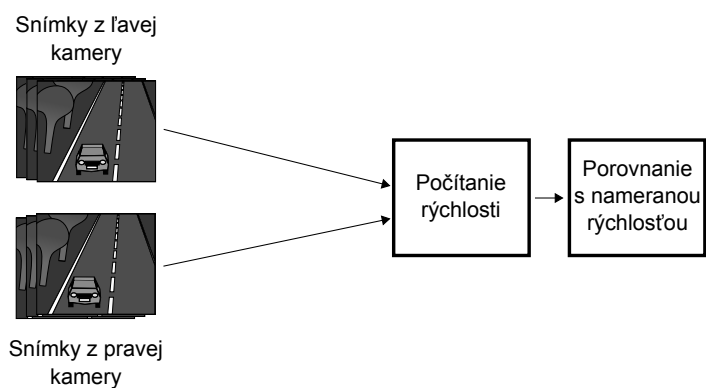
Následne bola otestovaná rýchlosť algoritmu na overenie, či by bol algoritmus schopný fungovať v reálnom čase. Pri testovaní bol program spustený paralelne pre obidve kamery. Program bol schopný spracovať sadu 98936 snímok za 8 minút a 26 sekúnd. Z toho 98905 snímok bolo spracovaných rýchlym algoritmom a 31 snímok oboma algoritmami. Program spracoval priemerne 195.5 snímok (97.8 stereosnímok) za sekundu. Táto rýchlosť je dostatočná pre spracovanie snímok.

6.3 Testovanie reálnej funkčnosti algoritmu

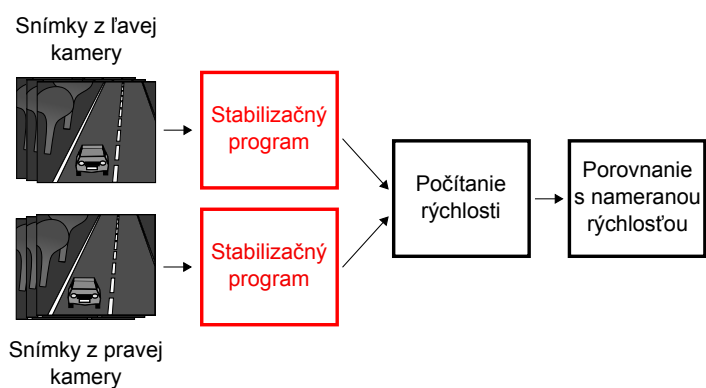
Záverečné testovanie bolo realizované na rovnakej sade stereosnímok, ale v spolupráci s Ing. Pavlom Najmanom, spoluautorom programu na meranie rýchlosti vozidiel pomocou stereokamery [16]. On pomocou svojho programu zo stereozáznamu spočítal rýchlosť vozidiel a porovnal ich s rýchlosťou vozidiel nameranou pomocou lidar. Odchýlky vypočítaných rýchlostí v kilometroch za hodinu oproti nameraným údajom sú uvedené v tabuľke 6.1 v stĺpci „Pôvodný záznam“. Tieto výsledky sú použité ako referenčné.

Vstupom môjho programu boli tie isté nestabilizované snímky, z ktorých boli poslednou verziou môjho programu vypočítané transformačné matice. Tieto matice Ing. Najman použil na transformovanie pôvodných vstupných snímok a následne ich použil na výpočet rýchlosti vozidiel. Schéma testovania je znázornená na obrázku 6.6. Výsledky porovnania rýchlostí vypočítaných zo stabilizovaných snímok voči nameraným rýchlostiam sú uvedené v tabuľke 6.1 v stĺpci „Stabilizovaný záznam“.

Testovanie pôvodného záznamu



Testovanie stabilizovaného záznamu



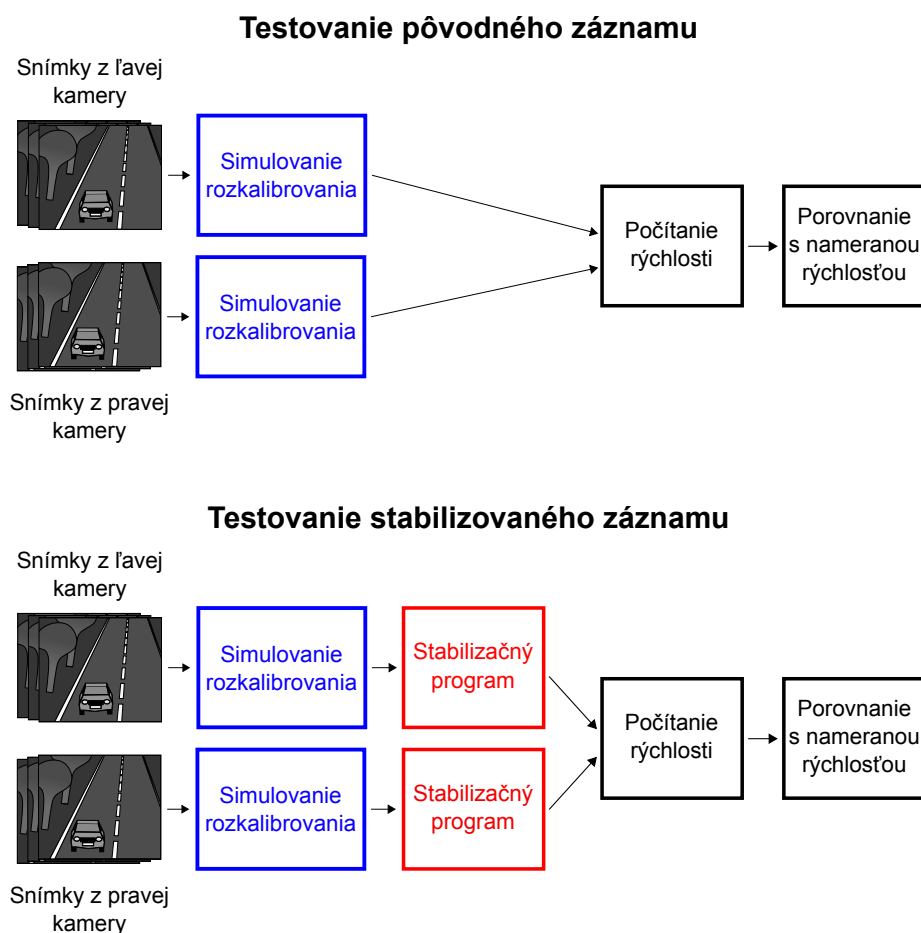
Obr. 6.6: Schéma testovania bez použitia a s použitím stabilizačného programu

	Chyba nameranej rýchlosti [km/h]	
	Pôvodný záznam	Stabilizovaný záznam
Stredná chyba	-0.053	-0.1144
Stredná kvadratická chyba	0.0402	0.0952
Smerodajná odchýlka	0.1934	0.2869
Stredná absolútna chyba	0.1591	0.2344
Maximálna absolútna chyba	0.7295	1.3908
Minimálna absolútna chyba	0.000 161	0.000 403
Maximálna chyba	0.7295	1.1804
Minimálna chyba	-0.5405	-1.3908

Tabuľka 6.1: Porovnanie výsledkov vypočítanej rýchlosti voči nameranej rýchlosti stabilizovaného a nestabilizovaného záznamu

Z jednotlivých výsledkov vyplýva, že stabilizácia nepriniesla spresnenie merania. Použité snímky ale boli zaznamenané tesne po kalibrácii a za krátku dobu snímania sa nestihli rozkalibrovať, takže referenčné výsledky boli dostatočne presné a nepotrebovali dodatočnú stabilizáciu. Jeden z predpokladov bol, že vypočítané matice majú v sebe šum, ktorý zhoršuje dostatočne presné výsledky. V ďalšom kroku som preto umelo posunul pôvodné snímky, čím som simuloval situáciu keď sa kamery stihnú rozkalibrovať. Všetky snímky z ľavej kamery boli posunuté o jeden pixel doprava a dole a všetky snímky ľavej kamery boli posunuté doľava a hore. Na nových umelo posunutých snímkach bol znovu spustený program na výpočet rýchlosti vozidiel. Keďže kamera bola skalibrovaná pre pôvodnú pozíciu, očakávaný výsledok bol, že chyba bude vyššia ako pri nestabilizovaných neposunutých snímkach. Výsledky tohto pokusu sa nachádzajú v tabuľke 6.2 v stĺpci „Posunutý pôvodný záznam“.

Ďalej boli tie isté posunuté snímky vstupom môjho programu. Ako referenčné snímky boli použité pôvodné neposunuté snímky. Výsledné matice boli použité na transformovanie posunutých snímkov a následne použité na výpočet rýchlosti vozidiel. Schéma tohto testu je znázornená na obrázku 6.7. Výsledky porovnania vypočítaných rýchlostí z tohto pokusu s nameranými rýchlosťami sú uvedené v tabuľke 6.2 v stĺpci „Posunutý stabilizovaný záznam“.



Obr. 6.7: Schéma testovania simulovaného rozkalibrovania kamery bez použitia a s použitím stabilizačného programu

	Chyba nameranej rýchlosti [km/h]	
	Posunutý pôvodný záznam	Posunutý stabilizovaný záznam
Stredná chyba	1.925	−0.196
Stredná kvadratická chyba	4.112	0.111
Smerodajná odchýlka	0.6382	0.2694
Stredná absolútna chyba	1.985	0.264
Maximálna absolútna chyba	3.518	1.458
Minimálna absolútna chyba	0.0719	0.0004
Maximálna chyba	3.518	1.010
Minimálna chyba	−2.190	−1.458

Tabuľka 6.2: Porovnanie výsledkov vypočítanej rýchlosti voči nameranej rýchlosti stabilizovaného a nestabilizovaného záznamu pri **simulovaní rozkalibrovania** kamery

V tomto prípade boli výsledky stabilizovaného záznamu výrazne lepšie. Stredná chyba bola takmer desaťnásobne nižšia a maximálna chyba bola menšia ako polovica maximálnej chyby nestabilizovaného záznamu. Pri simulácii rozkalibrovania systému môj program zlepšil presnosť vypočítaných rýchlostí.

Kapitola 7

Záver

Cieľom tejto práce bolo navrhnuť a implementovať program na zlepšenie presnosti meraní polohy a rýchlosti pomocou dopravnej stereokamery na základe kompenzácie nežiadúceho pohybu kamery. Cieľ práce bol splnený.

Preštudoval som dostupnú literatúru na tému metód stereospracovania obrazu a ich presnosti vrátane kalibračných postupov. Naštudované informácie sú zhrnuté v kapitole 2. V kapitole 3 som popísal algoritmy a postupy používané pri hľadaní transformácií medzi obrazmi. Na základe naštudovaných informácií som v kapitole 4 vybral parametre, ktoré by mala spĺňať a vlastnosti, ktoré by mala mať metóda spresnenia stereospracovania obrazu. V kapitole 5 je popísaný spôsob, ako bol daný program implementovaný. V kapitole 6 sú zhrnuté spôsoby, ktorými som môj program testoval a aké výsledky boli dosiahnuté.

Program analyzuje pohyb snímok voči referenčnej snímke. Ľavá a pravá snímka stereosnímkou sú spracované nezávisle. Na zistenie pohybu je použitá poloha rohových bodov na snímkach voči ich polohe na referenčnej snímke. Na nájdenie rohových bodov je použitý detektor Shi-Tomasi a subpixelový detektor rohov pracujúci na princípe kolmých vektorov. Rohové body sú porovnávané pomocou normalizovanej vzájomnej korelácie. Výstupom programu sú transformačné matice popisujúce afinnú transformáciu kompenzujúcu pohyb snímok voči referenčnej snímke.

Najskôr bol algoritmus overený na snímkach z čerstvo skalibrovaného systému, kde stabilizácia snímok nepriniesla spresnenie výsledkov oproti rovnakému vyhodnoteniu výsledkov na nestabilizovaných snímkach. Naopak došlo, pravdepodobne vplyvom malého vneseného šumu, k marginálnemu zhoršeniu výsledkov. Ďalej som sa zameral na účinnosť implementovaného algoritmu pri väčších pohyboch kamier. Z pôvodnej vzorky snímok bola vytvorená vzorka s umelo simulovanou rozkalibrovanou stereokamerou, teda snímky s väčším posunom. V tomto prípade mal môj program výrazne pozitívny vplyv na presnosť určenia rýchlosti.

Z výsledkov získaných testovaním programu na simulovaných snímkach sa dá predpokladať, že by môj program zlepšil presnosť meraní, keby bol nasadený na reálny systém určený na meranie rýchlosti pracujúci nepretržite niekoľko mesiacov alebo rokov, ktorý by sa postupom času rozkalibroval.

V budúcnosti by bolo vhodné otestovať program na väčšej sade reálnych snímok, ktorá by obsahovala dlhšie časové úseky, rôzne typy počasia ako napríklad dážď a sneh, a rôzne časti dňa. Taktiež by v budúcnosti bolo vhodné zamerať sa na spomínaný šum, ktorý môj program vytvorí. Ten pravdepodobne spôsobuje použitá metóda hľadania subpixelovej pozície rohov. Tá by mohla byť nahradená napríklad hľadaním najväčšej vzájomnej korelácie medzi výrezmi rohov prevzorkovanými na vyššie rozlíšenie. Ďalším vylepšením by mohlo byť

použitie pozície rohov z predchádzajúcej snímky pri dohľadávaní rohov, a nie pozície rohov z referenčnej snímky. Dá sa totiž predpokladať, že pozície rohov z predchádzajúcej snímky budú bližšie k pozíciám rohov na aktuálnej snímke ako k pozíciám rohov na referenčnej snímke. V budúcnosti by bolo vhodné prepísať program ako knižnicu tak, aby stabilizácia mohla byť použitá priamo v programoch na meranie rýchlosti pomocou stereokamery.

Literatúra

- [1] BEBIS, G. *The Geometry of Perspective Projection* [online]. CSE Department, Penn State University, 2004 [cit. 2021-04-11]. Dostupné z: <https://www.cse.unr.edu/~bebis/CS791E/Notes/PerspectiveProjection.pdf>.
- [2] CHU, C.-H. Video stabilization for stereoscopic 3D on 3D mobile devices. In: *2014 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2014, s. 1–6. DOI: 10.1109/ICME.2014.6890150.
- [3] COLLINS, R. *Lecture 7: Correspondence Matching* [online]. CSE Department, Penn State University, 2007 [cit. 2021-04-12]. Dostupné z: <http://www.cse.psu.edu/~rtc12/CSE486/lecture07.pdf>.
- [4] DUBROFSKY, E. *Homography Estimation*. Vancouver, Canada, 2009. Diplomová práca. The University of British Columbia.
- [5] EL BOUZIADY, A., THAMI, R. O. H., GHOGHO, M., BOURJA, O. a EL FKIHI, S. Vehicle speed estimation using extracted SURF features from stereo images. In: *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*. 2018, s. 1–6. DOI: 10.1109/ISACV.2018.8354040.
- [6] FERNÁNDEZ LLORCA, D., SALINAS, C., JIMENEZ, M., MORCILLO, A., IZQUIERDO, R. et al. Two-camera based accurate vehicle speed measurement using average speed at a fixed point. In: November 2016, s. 2533–2538. DOI: 10.1109/ITSC.2016.7795963.
- [7] FORSYTH, D. *Computer vision : a modern approach*. 2. vyd. Boston, USA: Pearson, 2012. ISBN 978-0-13-608592-8.
- [8] GONZALEZ, R. C. a WOODS, R. E. *Digital image processing*. 3. vyd. Prentice Hall, 2008. ISBN 978-0-13-335672-4.
- [9] HABIB, A., DETCHEV, I. a KWAK, E. Stability Analysis for a Multi-Camera Photogrammetric System. *Sensors (Basel, Switzerland)*. August 2014, zv. 14, č. 8, s. 15084–112. DOI: 10.3390/s140815084. ISSN 1424-8220.
- [10] HISHAM, M. B., YAAKOB, S. N., RAO, R. A. A., NAZREN, A. B. A. a WAFI, N. M. Template Matching using Sum of Squared Difference and Normalized Cross Correlation. In: *2015 IEEE Student Conference on Research and Development (SCOREd)*. IEEE, 2015, s. 100–104. DOI: 10.1109/SCORED.2015.7449303.
- [11] HUANG, T. Computer Vision: Evolution And Promise. In: CERN. *1996 CERN School of Computing: Egmond aan Zee, The Netherlands 8 - 21 Sep 1996*. 19th CERN School of Computing. Geneva: CERN, 1996, s. 46–49. DOI: 10.5170/CERN-1996-008.21. ISBN 9789290830955.

- [12] JALALAT, M., NEJATI, M. a MAJIDI, A. Vehicle detection and speed estimation using cascade classifier and sub-pixel stereo matching. In: *2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)*. 2016, s. 1–5. DOI: 10.1109/ICSPIS.2016.7869890.
- [13] KEYS, R. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1981, zv. 29, č. 6, s. 1153–1160. DOI: 10.1109/TASSP.1981.1163711.
- [14] LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. november 2004, zv. 60, č. 2, s. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94. ISSN 0920-5691.
- [15] MORAVEC, H. P. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. 1980. Dizertačná práca. Computer Science Department, Stanford University. Dostupné z: https://kilthub.cmu.edu/articles/journal_contribution/Obstacle_avoidance_and_navigation_in_the_real_worldby_a_seeing_robot_rover/6557033/1.
- [16] NAJMAN, P. a ZEMČÍK, P. Vehicle Speed Measurement Using Stereo Camera Pair. *IEEE Transactions on Intelligent Transportation Systems*. 2020, s. 1–9. DOI: 10.1109/TITS.2020.3035262. ISSN 1558-0016.
- [17] OPENCV. *OpenCV – Harris Corner Detection* [online]. 2015 [cit. 2021-04-07]. Dostupné z: https://docs.opencv.org/3.0.0/dc/d0d/tutorial_py_features_harris.html.
- [18] OPENCV. *OpenCV – Understanding Features* [online]. 2015 [cit. 2021-04-06]. Dostupné z: https://docs.opencv.org/3.0.0/df/d54/tutorial_py_features_meaning.html.
- [19] ORAM, D. Rectification for Any Epipolar Geometry. *Proc. British Machine Vision Conf.* Apríl 2002. DOI: 10.5244/C.15.67.
- [20] PAN, B., SHI, W. a LUBINEAU, G. Effect of camera temperature variations on stereo-digital image correlation measurements. *Applied Optics*. The Optical Society. 2015, zv. 54, č. 34, s. 10089–10095. DOI: 10.1364/AO.54.010089. ISSN 0003-6935.
- [21] ROSTAM AFFENDI HAMZAH, ROSMAN ABD RAHIM a ZARINA MOHD NOH. Sum of Absolute Differences algorithm in stereo correspondence problem for stereo matching in computer vision application. In: *2010 3rd International Conference on Computer Science and Information Technology*. IEEE, 2010, sv. 1, s. 652–657. DOI: 10.1109/ICCSIT.2010.5565062.
- [22] RUTANEN, K., GÓMEZ HERRERO, G., ERIKSSON, S.-L. a EGIAZARIAN, K. Least-Squares Transformations between Point-Sets. In: KÄMÄRÄINEN, J.-K. a KOSKELA, M., ed. *Image Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, s. 501–511. ISBN 978-3-642-38886-6.
- [23] SHI, J. a TOMASI. Good features to track. In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1994, s. 593–600. DOI: 10.1109/CVPR.1994.323794.

- [24] SOCHOR, J., JURÁNEK, R. a HEROUT, A. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. *Computer Vision and Image Understanding*. 2017, zv. 161, s. 87–98. DOI: <https://doi.org/10.1016/j.cviu.2017.05.015>. ISSN 1077-3142.
- [25] TYMCHYSHYN, V. a KHLEVNIUK, A. *Beginner's guide to mapping simplexes affinely*. Apríl 2019. DOI: 10.13140/RG.2.2.13787.41762.
- [26] VILLENA MARTÍNEZ, V., FUSTER GUILLÓ, A., AZORÍN LÓPEZ, J., SAVAL CALVO, M., MORA PASCUAL, J. et al. A Quantitative Comparison of Calibration Methods for RGB-D Sensors Using Different Technologies. *Sensors*. 2017, zv. 17, č. 2. DOI: 10.3390/s17020243. ISSN 1424-8220.
- [27] WIKIPEDIA CONTRIBUTORS. *Corner detection* — *Wikipedia, The Free Encyclopedia* [online]. 2021 [cit. 2021-04-07]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Corner_detection&oldid=1013223046.
- [28] XIANG, W., CONLY, C., MCMURROUGH, C. D. a ATHITSOS, V. A Review and Quantitative Comparison of Methods for Kinect Calibration. In: New York, NY, USA: Association for Computing Machinery, 2015. IWOAR '15. DOI: 10.1145/2790044.2790056. ISBN 9781450334549.
- [29] YANG, L., LI, M., SONG, X., XIONG, Z., HOU, C. et al. Vehicle Speed Measurement Based on Binocular Stereovision System. *IEEE Access*. 2019, zv. 7, s. 106628–106641. DOI: 10.1109/ACCESS.2019.2932120.
- [30] ZHANG, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000, zv. 22, č. 11, s. 1330–1334. DOI: 10.1109/34.888718.
- [31] ZHU, W., MA, C., XIA, L. a LI, X. A Fast and Accurate Algorithm for Chessboard Corner Detection. In: *2009 2nd International Congress on Image and Signal Processing*. IEEE, Oct 2009, s. 1–5. DOI: 10.1109/cisp.2009.5304332.